Do Data Warehouse à Narrativa: Uma Arquitetura HTAP para Análise de Dados de Criminalidade*

Ryan Serra[1], Daniel Ribeiro[1], Arben H. S. Tafili[1], Paulo S. Costa-Jr.[1], João V. Silva-Leite[1], Rodrigo S. Monteiro[1], Daniel de Oliveira[1], Marcos Bedo[1]

[1]Instituto de Computação – Universidade Federal Fluminense – IC/UFF
Niterói, RJ – Brazil

{ryanserrasilva, danielrr, arbensilva, joaovitorleite}@id.uff.br,

paulorscj@id.uff.br, {salvador, danielcmo@ic, marcosbedo}@ic.uff.br

Abstract. *This study proposes the* `CrimeSmartScribe` *platform for crime data analysis, supporting continuous data ingestion and interaction through natural language queries and interactive dashboards. The approach is grounded in a multidimensional logical model deployed on a hybrid environment (HTAP) that leverages (i) virtual views, (ii) Retrieval-Augmented Generation strategies, and (iii) an agentic architecture for text-to-SQL translation, to generate journalistic-style responses with infographics for user-driven ad-hoc questions. The proposed solution was evaluated on a real-world US crime dataset, achieving operational robustness, improved text-to-SQL translation, and the platform achieved high usability scores among specialists and non-specialist users.*

Resumo. *Este artigo propõe a plataforma* `CrimeSmartScribe` *para análise de dados de criminalidade, com ingestão contínua e suporte a consultas em linguagem natural e dashboards interativos. A abordagem baseia-se em um modelo lógico multidimensional em ambiente híbrido transacional/analítico (HTAP) e emprega (i) visões virtuais, (ii) estratégias de geração aumentada de recuperação (RAG) e (iii) uma arquitetura multiagente para tradução texto-para-SQL e geração das respostas em tons jornalísticos com infográficos. Avaliada em dados criminais norte-americanos, a solução demonstrou robustez operacional, melhoria na tradução texto-para-SQL com o uso de RAG e alcançou altos índices de usabilidade entre usuários especialistas e não especialistas.*

1. Introdução

A proliferação de múltiplas fontes de informação tornou mais complexa a obtenção de dados criminais confiáveis, contextualizados e acessíveis de forma centralizada, uma vez que o processo de curadoria e qualidade dos dados é diferente para cada fonte de informação [Greenspan et al. 2024]. Ainda que agências tornem públicos registros de criminalidade, a transparência informacional não se limita ao acesso aos dados, exigindo processos interpretativos que permitam a transformação desses registros em conhecimento socialmente relevante. Essa limitação de acesso decorre da barreira tipicamente

*Os autores agradecem à Fundação Carlos Chagas Filho de Amparo à Pesquisa do Estado do Rio de Janeiro pelo apoio financeiro durante este estudo (E-26/204.238/2024 e E-26/204.544/2024).

imposta pelos fluxos de análise de dados, que requer competências especializadas para o tratamento, integração e análise de grandes volumes de dados, incluindo processos complexos de ETL (*Extract, Transform, and Load*), modelagem multidimensional e domínio de linguagens de consulta estruturadas, como SQL ou afins. Consequentemente, uma parcela expressiva de dados com potencial informativo para políticas públicas e análises sociais permanece inacessível à maioria dos atores sociais, devido à exigência de conhecimentos especializados em Engenharia de Dados [Silva et al. 2022, Mtchedlidze 2024].

O desafio em acessibilidade. A disputa de narrativas sobre criminalidade e segurança pública está associada, dentre outros, a dois fatores: a ausência de análises estruturadas de dados e a restrição do acesso aos dados criminais a usuários com formação técnica especializada [Greenspan et al. 2024]. Assim, a apropriação dessas informações torna-se limitada para tomadores de decisão, formadores de opinião, jornalistas, gestores públicos e para o público não especializado em geral. Além disso, a maior parte das soluções existentes concentra-se exclusivamente em ambientes de Processamento Analítico Online (OLAP) com a visualização dos dados por meio de *dashboards* tradicionais, com indicadores-chave de desempenho (KPIs) pré-definidos [Hidalgo and Centeno 2023]. Essa análise rígida, além de não contemplar métricas não codificadas, carece da flexibilidade necessária para responder a perguntas *ad-hoc* e exploratórias que surgem durante uma investigação [Diakopoulos 2019, Jamil and Rubaiat 2024, Maltezos et al. 2025]. Por exemplo, a identificação de uma redução de 20% nos registros de assaltos em vias públicas é trivial em um gráfico de pizza; a correlação desse resultado, contudo, exige um processo *inteligente* de análise, capaz de cruzar tais dados a informações contextuais transacionais (OLTP), como decretos de isolamento sanitário no período da redução observada.

LLMs e IA generativa. A adoção de Grandes Modelos de Linguagem (LLMs) ampliou as possibilidades de automação de consultas por meio de linguagem natural. Esses modelos possibilitam a tradução de consultas em linguagem natural para representações formais, como SQL (*text-para-sql*), e a integração automatizada de dados provenientes de múltiplas fontes. Observa-se, contudo, um desafio relevante na aplicação dessas soluções à análise de dados criminais, decorrente do caráter predominantemente proprietário das abordagens existentes e de sua inadequação a fluxos analíticos de natureza jornalística, os quais demandam não apenas a consulta aos dados, mas sobretudo sua interpretação contextualizada e a comunicação direta e compreensível de resultados para públicos não especializados [Siqueira et al. 2025, Chien et al. 2025]. Assim, a inexistência de uma abordagem sistemática limita o uso dessas tecnologias na análise de dados criminais, já que apenas conectar um LLM a um ambiente OLAP não seria suficiente para construir narrativas orientadas a dados, potencialmente incorrendo em diversas "alucinações" do modelo com relação ao domínio. Mitigar esse problema requer modelagem lógica cuidadosa e mecanismos de contenção e contextualização, como exemplificação e geração aumentada de recuperação (RAG) [Fürst et al. 2024, Fan et al. 2024, Pan et al. 2024].

Para superar esses desafios, a proposta `CrimeSmartScribe` fundamenta-se em um modelo lógico multidimensional instanciável em diferentes sistemas NewSQL, viabilizando um ambiente de processamento híbrido transacional/analítico (HTAP) que integra dados analíticos, da aplicação, além de dados vetoriais necessários à execução das consultas. A ingestão é realizada por meio de um pipeline elástico de ETL baseado em Apache Spark, enquanto os dados vetoriais decorrem de uma base de perguntas e res-

postas para suportar estratégias RAG no tratamento de consultas em linguagem natural e na tradução texto-para-SQL. Com o objetivo de aprimorar o desempenho do processo de busca e interpretação das consultas, são investigadas três estratégias complementares: *(i)* visões virtualizadas, que simplificam o esquema lógico exposto aos modelos de linguagem sem comprometer a eficiência do modelo físico; *(ii)* o uso de exemplos contextuais via RAG; e *(iii)* uma arquitetura multiagentes, responsável pela composição contextual, tradução texto-para-SQL, geração das respostas, produção de infográficos e edição final.

Para avaliação experimental e como estudo de caso, foi utilizada uma base de dados pública americana contendo registros de crimes e fatalidades. Os resultados indicam que *(i)* a infraestrutura HTAP proposta apresentou robustez tanto para ingestão quanto para consulta em escala, mesmo sob condições híbridas; *(ii)* o uso combinado de visões virtualizadas e RAG melhorou os resultados da tradução texto-para-SQL, evidenciando que visões virtualizadas atuam como uma camada semântica eficaz, frequentemente produzindo ganhos mais consistentes do que o simples aumento de contexto textual; e *(iii)* respostas apresentadas em tom jornalístico mostraram-se mais intuitivas e preferíveis ao público geral quando comparadas a *dashboards* interativos tradicionais, ampliando a usabilidade e o acesso à informação por usuários não especializados.

Contribuições. Para investigar os desafios de acessibilidade e qualidade na exploração de dados criminais por linguagem natural, este trabalho apresenta as seguintes contribuições: *(i)* propor uma plataforma HTAP unificada baseada em NewSQL, capaz de suportar simultaneamente cargas OLTP, OLAP e consultas vetoriais associadas a RAG sem a necessidade de camadas intermediárias especializadas; *(ii)* apresentar uma avaliação quantitativa do custo-benefício entre estratégias *zero-shot*, *few-shot* e RAG aplicadas à tradução texto-para-SQL, com e sem o uso de visões virtualizadas, em um cenário de banco de dados real; e *(iii)* integrar uma arquitetura multiagente voltada não apenas à geração de consultas e análises, mas também à produção de narrativas explicativas e visualizações analíticas. Essas contribuições exploram sinergias entre técnicas consolidadas de bancos de dados e abordagens recentes de IA generativa em um ambiente analítico unificado.

O restante deste artigo está organizado da seguinte forma. A Seção 2 discute os trabalhos relacionados; a Seção 3 descreve a metodologia proposta; a Seção 4 apresenta a avaliação experimental; e a Seção 5 discute as considerações finais e trabalhos futuros.

2. Fundamentos e Trabalhos Relacionados

O acesso inteligível a dados complexos constitui um desafio estado-da-arte em Engenharia de Dados, incluindo o gerenciamento eficiente de ambientes HTAP, tradução *texto-para-SQL* e o emprego de RAG em arquiteturas multiagentes. Na sequência, revisam-se as lacunas identificadas e as estratégias propostas na literatura para abordá-las.

Modelagem multidimensional. O modelo multidimensional provê abstrações lógicas de alto nível para a organização dos dados (*e.g.*, cubos) em ambientes OLAP, enquanto a linguagem SQL viabiliza a especificação dessas estruturas e de seus operadores por meio de expressões declarativas de alto nível. No plano físico, os dados podem ser organizados segundo diferentes modelos de armazenamento, escala e redundância. Trabalhos relacionados indicam que a execução de cargas OLAP em NoSQL por meio de consultas SQL é operacionalmente viável, sendo que a aplicação da mesma carga resulta em variações significativas de desempenho entre categorias

NoSQL [Mohan et al. 2024]. A migração de estruturas entre ambientes NoSQL, no entanto, é não trivial, evidenciando o acoplamento com o modelo físico que precisa ser escolhido de antemão [García-Gil et al. 2017, Mohan et al. 2024]. Essa limitação é superada em ambientes NewSQL-like, que preservam a independência do modelo multidimensional com suporte nativo a SQL [Taft et al. 2020, Chen et al. 2025]. Resultados experimentais indicam que esses ambientes apresentam alto desempenho para cargas OLTP e OLAP, com variações atribuídas aos respectivos modelos físicos. Sistemas NewSQL podem ser intercambiados sem impacto no esquema lógico [Siqueira et al. 2025].

Tradução *texto-para-SQL*. Traduções *texto-para-SQL* permitem que perguntas em linguagem natural sejam convertidas em instruções SQL equivalentes. Diferentes LLMs têm sido usadas para essa tarefa em benchmarks considerando cenários *zero-shot* e *few-shot* [Fürst et al. 2024, Luo et al. 2025, Shi et al. 2025]. Não obstante, avaliações em bancos de dados do mundo real evidenciam degradação de desempenho, associada a esquemas extensos, nomenclaturas e relacionamentos implícitos [Fürst et al. 2024, Shi et al. 2025]. O estudo de [Coelho et al. 2024] argumenta que essa limitação pode ser mitigada por meio da introdução de visões virtuais como camada semântica intermediária. Os resultados indicam que visões cuidadosamente projetadas reduzem a complexidade do esquema exposto ao modelo e melhoram significativamente a acurácia da tradução, sinalizando que o desempenho dos LLMs pode ser expressivamente melhorado por abstrações do modelo lógico realizadas diretamente em SQL sem qualquer impacto no modelo físico.

RAG. A geração aumentada por recuperação permite aos LLMs situarem adequadamente o contexto para a geração de texto. Trabalhos relacionados mostram que RAG, na forma de recuperação de exemplos de perguntas similares, melhora o desempenho dos LLMs em tarefas *texto-para-SQL* [Fan et al. 2024]. A operacionalização envolve *(i)* a definição e manutenção de uma base de pares pergunta–resposta representativos do domínio, *(ii)* a representação dessas perguntas em um espaço multidimensional (*embeddings*), e *(iii)* a indexação desses *embeddings* para a realização eficiente de buscas por similaridade. Essa base semântica atua como uma camada adicional de contexto com propriedades transacionais: requer atualização incremental, controle de consistência com o esquema subjacente e manutenção para acompanhar a evolução do domínio [Pan et al. 2024].

Sistemas multiagentes. Além da tradução *texto-para-SQL*, sessões de LLMs podem simular sistemas multiagentes de IA, nas quais tarefas são atribuídas a agentes especializados que interagem entre si para a resolução de desafios. A definição de *papéis* aos agentes permite isolar etapas, reduzindo a propagação de erros ao longo do processo [Ferber and Weiss 1999, Miehling et al. 2025]. Diversos *workflows* de colaboração são encontrados na literatura, incluindo arquiteturas de auto-colaboração inspiradas em equipes humanas, onde agentes produzem e validam artefatos de forma incremental [Dong et al. 2024]. Um exemplo dessa colaboração é a observada em produções jornalísticas, nas quais a elaboração de conteúdos envolve papéis distintos, como repórteres responsáveis pela coleta de informações, editores encarregados da curadoria editorial e especialistas dedicados à visualização. Embora essa divisão de responsabilidades já aconteça no jornalismo, a simulação desse fluxo colaborativo por agentes de IA sobre ambientes OLAP permanece um desafio em aberto [Silva et al. 2022, Mtchedlidze 2024].

Ambientes HTAP. O modelo multidimensional permite a representação lógica de dados analíticos de forma independente do modelo físico, mantendo interoperabilidade por

meio de consultas SQL. Nesse modelo, visões podem ser utilizadas como uma primeira camada semântica, reduzindo a complexidade do esquema exposto e apoiando a tradução *texto-para-SQL*. Complementarmente, uma segunda camada semântica pode ser estabelecida por meio de estratégias de RAG, incorporando exemplos históricos e conhecimento contextual ao processo de tradução. Do ponto de vista do gerenciamento de dados, esse arranjo resulta em um ambiente heterogêneo acessado por agentes de IA, que combina diferentes perfis de carga e representação: *(i)* consultas OLAP sobre dados analíticos curados, *(ii)* operações OLTP sobre bases de perguntas–respostas e metadados operacionais, bem como *(i)* estruturas multidimensionais relacionais e *(ii)* dados vetoriais associados a *embeddings*. A coexistência desses requisitos motiva o uso de sistemas capazes de suportar, de forma integrada, cargas transacionais e analíticas.

Ambientes HTAP com suporte nativo a SQL preservam a estrutura lógica, ao mesmo tempo em que exploram execução distribuída de cargas OLTP e OLAP, eliminando a necessidade de camadas intermediárias entre sistemas especializados. Parte da literatura aborda essa convergência por meio de arquiteturas *lakebase*, baseadas em armazenamento compartilhado para suportar múltiplos mecanismos de execução [Zaharia 2025]. Em contraste, abordagens NewSQL integram processamento transacional e analítico sob o modelo relacional, mantendo a independência entre o esquema lógico e as decisões físicas de particionamento e replicação. Nesse contexto, sistemas como Spanner e CockroachDB [Taft et al. 2020, Siqueira et al. 2025] permitem sustentar, de forma unificada, modelos multidimensionais, camadas semânticas via visões, tradução *texto-para-SQL* e estratégias de RAG, ao oferecer acesso uniforme a dados transacionais, analíticos e vetoriais em um único ambiente lógico.

Trabalhos relacionados. No domínio de dados criminais e do jornalismo orientado a dados, iniciativas existentes concentram-se majoritariamente em plataformas analíticas centralizadas e mecanismos de *busca assistida*, com ênfase na exploração e visualização de grandes volumes de registros públicos [Diakopoulos 2019, Jamil and Rubaiat 2024, Maltezos et al. 2025]. Em paralelo, sistemas OLAP têm sido amplamente empregados para suportar análises estatísticas e *dashboards* interativos sobre criminalidade, conforme documentado em estudos empíricos voltados a políticas públicas [Mtchedlidze 2024, Hidalgo and Centeno 2023]. Essas abordagens, contudo, tendem a apresentar limitações: ou operam sobre dados textuais não estruturados, sem suporte nativo a consultas analíticas estruturadas, ou dependem de interfaces rígidas baseadas em *dashboards*. Paralelamente, arquiteturas baseadas em agentes e fluxos colaborativos vêm sendo investigadas como forma de apoiar processos jornalísticos complexos, ainda que, em geral, sem integração direta a ambientes analíticos estruturados [Maltezos et al. 2025, Chien et al. 2025].

De forma geral, os trabalhos analisados evidenciam avanços relevantes na ampliação do acesso a dados por meio de abstrações analíticas, mecanismos de busca assistida e automação baseada em IA. Entretanto, a literatura aborda esses desafios de maneira predominantemente fragmentada, tratando isoladamente estratégias de tradução *texto-para-SQL*, arquiteturas analíticas ou fluxos colaborativos baseados em agentes. A proposta do `CrimeSmartScribe`, apresentada a seguir, integra sistematicamente essas frentes ao combinar modelo multidimensional, camadas semânticas baseadas em visões, estratégias de RAG, execução HTAP e orquestração multiagente, viabilizando a geração e exploração de narrativas orientadas a dados em um ambiente unificado.

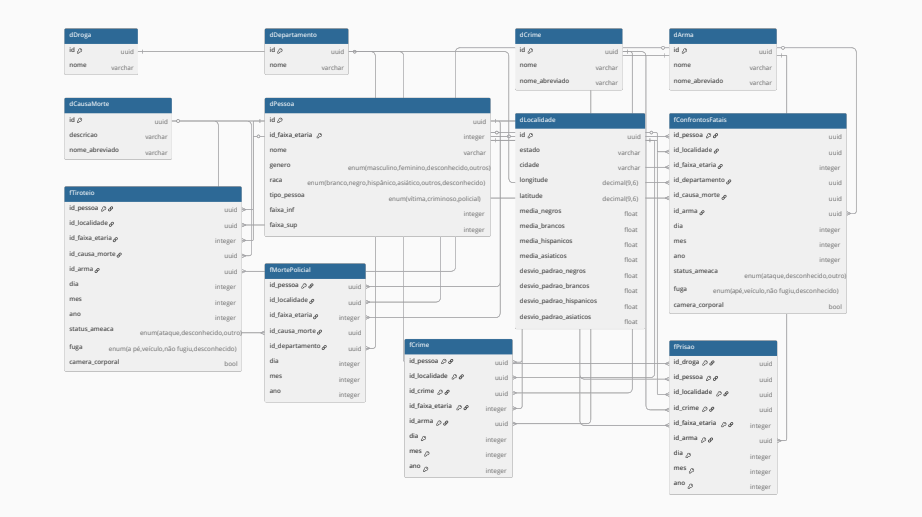


Figura 1. Modelo lógico *fact constellation* simplificado do `CrimeSmartScribe`.

3. Materiais e Métodos

A plataforma `CrimeSmartScribe` foi projetada como uma infraestrutura integrada para ingestão, organização e análise de dados criminais. Esses dados podem vir de múltiplas fontes e, para uma análise numericamente mais precisa e de maior qualidade, os dados são conformados em um modelo lógico após sua transformação.

Pipeline de ingestão. Os dados podem ser extraídos de arquivos .csv ou de APIs de repositórios públicos. Na implementação realizada, os dados são armazenados em uma área de *staging*, que pode ser configurada de forma dinâmica, incluindo MinIO, Digital Ocean Spaces ou Google Drive. Devido aos recursos de infra-estrutura disponíveis, optou-se pelo desenvolvimento customizável de um *notebook* python ambiente *Jupyter* para realizar o processo de ETL. Sempre que possível, as transformações nesse *notebook* foram realizadas com comandos SQL-like, para permitir as seguintes vantagens: *(i)* caso exista um *kernel* adequadamente configurado, as transformações podem ser executadas com eficiência via *Apache Spark* (para arquivos .csv) ou *Apache Flink* (para *streaming* de APIs), apenas trocando a engine de execução; e *(ii)* as transformações podem ser isoladas, podendo ser executadas como um *script* diretamente disparado tanto em arquitetura serverless quanto com o apoio de orquestradores como *Apache Airflow*. As transformações realizadas podem ser categorizadas entre: *(i)* normalização de esquema, com seleção de atributos relevantes e exclusão de metadados administrativos; *(ii)* limpeza e padronização textual, incluindo remoção de caracteres especiais e uniformização; *(iii)* codificação categórica de atributos textuais de domínio livre (*e.g.*, etnia e tipo de arma); *(iv)* reforço de tipagem, com conversão explícita dos tipos de dados; e *(v)* decomposição temporal, transformando atributos para suporte a operações OLAP.

Modelo Lógico. Na plataforma, os dados são inseridos em um *Data Warehouse* distribuído, fundamentado em um modelo *Fact Constellation* – Ver Figura 1. As dimensões desnormalizadas permitem realizar a filtragem e fornecem o contexto descritivo temporal, geográfico e categórico para os eventos criminais armazenados nas tabelas fato. Essa modelagem é relevante para permitir a construção de respostas contextualizadas e com cruzamento de contexto, como distribuição étnica associada a tiroteios, ou distribuição de

pirâmide etária por localidade associada a confrontos fatais, crimes e prisões. Essa modelagem reflete uma decisão de projeto que não está livre de *trade-offs*, já que a coexistência de múltiplas tabelas fato semanticamente relacionadas pode introduzir sobreposição conceitual entre eventos, exigindo cuidado na interpretação dos resultados analíticos. A proposta para o modelo é manter essa capacidade de responder a consultas exploratórias complexas em detrimento de uma modelagem mais estrita, como um modelo *Star*.

Infraestrutura HTAP. Enquanto o modelo lógico permite capturar níveis de relacionamento entre fato e dimensões, a fase de ETL implementada possui capacidade elástica de processamento e ingestão. Assim, foi escolhido um SGBD NewSQL para a instanciação do modelo e hospedar as demais bases necessárias ao funcionamento da plataforma, incluindo a base de RAG e a base de controle de sessões e agentes. Devido à compatibilidade com o protocolo *wire* do PostgreSQL (o que facilita o *benchmarking* comparativo com ambientes centralizados) e ao suporte nativo à linguagem SQL declarativa, bem como a mecanismos de indexação para a execução eficiente de consultas por similaridade utilizadas em estratégias de RAG, optou-se pelo uso do *CockroachDB*. Essa escolha atende aos requisitos de ambientes HTAP adotados ao longo da arquitetura, embora sistemas funcionalmente análogos, como o Spanner, constituam alternativas viáveis.

Visões Virtuais e Construção da Base de Perguntas RAG. A camada RAG foi projetada para fornecer contexto semântico via similaridade para melhorar o processo de tradução *texto-para-SQL*. Para isso foi construída uma base de perguntas e respostas, que seguiu uma metodologia incremental *snowball*. Inicialmente, foi definido um conjunto reduzido de consultas, elaboradas manualmente a partir de consultas e padrões analíticos recorrentes observados em perguntas de usuários genéricos durante explorações preliminares do repositório de dados. Foi observado que, predominantemente, essas consultas priorizaram operações de agregação, filtragem e ordenação sobre dimensões temporais e geográficas. A partir desse núcleo, novas perguntas foram adicionadas de forma iterativa, por meio de especialização (*e.g.*, refinamento de granularidade temporal e espacial) e generalização (*e.g.*, substituição de dimensões e métricas), ampliando progressivamente a cobertura semântica da base. Em seguida, foram analisados os planos de consulta no ambiente HTAP para eliminar consultas que geravam planos muito similares e outra expansão da base para capturar variações relevantes na composição das perguntas da base. O conjunto resultante inclui, dentre outras, as seguintes consultas com operadores multidimensionais: *(i) Qual mês de maior incidência de roubos na proporção com o mês anterior para um intervalo de datas considerando os locais (rollup)? (ii) Qual mês de maior incidência de mortes por ações policiais considerando diferentes locais (rollup)?*

Os resultados dessa etapa também foram considerados para a construção das *visões virtuais* que não apenas tiveram o objetivo de *(i)* abstrair logicamente o modelo multidimensional para uma compreensão mais simplificada e já agregada do esquema, mas também *(ii)* identificar potenciais gargalos e ajustes de desempenho. Como exemplo de construção desse tipo de visão, destaca-se a visão para o fato de "confrontos fatais", que permite responder consultas relacionadas a óbitos em confrontos com forças de segurança de acordo com diferentes níveis de ameaça, perseguição, pirâmide etária, dentre outros. As visões do `CrimeSmartScribe` foram definidas manualmente com base em padrões analíticos recorrentes observados durante a construção incremental da base RAG, abstraindo junções e agregações frequentes do modelo multidimensional em um esquema mais

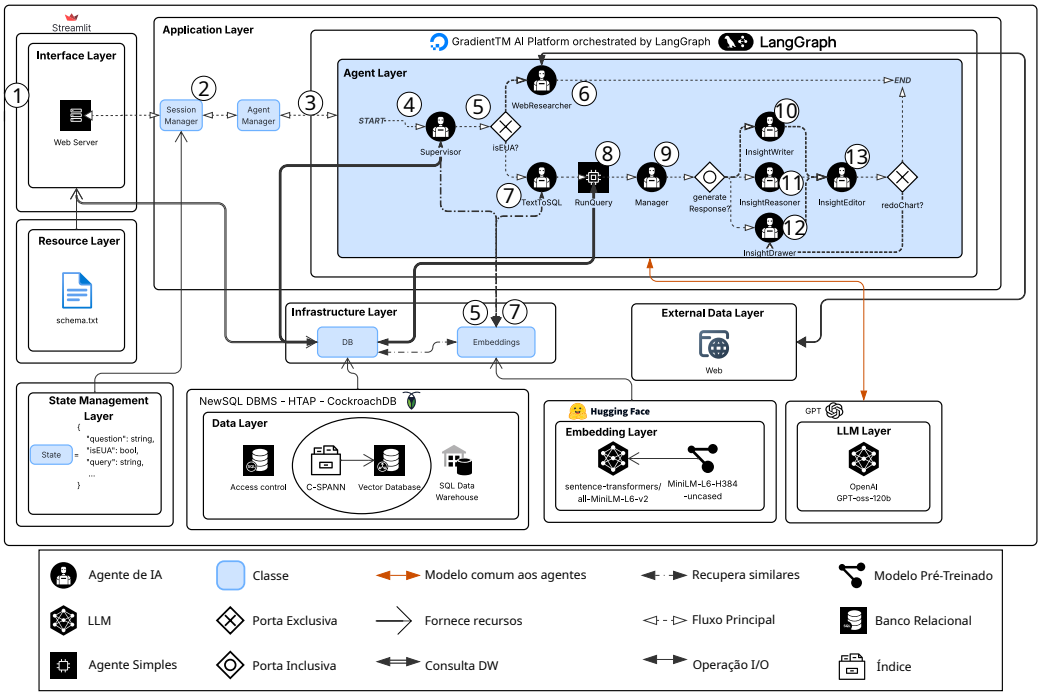


Figura 2. Visão geral da arquitetura da plataforma `CrimeSmartScribe`.

simples para consumo pela LLM. Após adicionar visões ao modelo lógico, as perguntas resultantes dessa segunda interação foram transformadas em uma representação vetorial (*embedding*) por meio de diferentes modelos de linguagem. Devido ao desempenho e a possibilidade de portar a solução para dentro da plataforma `CrimeSmartScribe` como um componente containerizado de baixo custo de infra-estrutura em nuvem, optou-se pela utilização do modelo de código aberto *sentence-transformers/all-MiniLM-L6-v2* disponibilizada no repositório *HuggingFace*. Esse modelo permite representar as perguntas em um espaço de 384-dimensões, inviabilizando buscas sequenciais durante o processo de RAG. Para superar esse desafio, foi empregada a indexação vetorial com o índice C-SCAN (*Clustered Scan*)[1] para permitir a execução distribuída de consultas *k*-NN *exatas*. Além disso, extensões da linguagem SQL no ambiente NewSQL viabilizam a composição de predicados de similaridade com relacionais convencionais em uma única consulta declarativa, sendo os operadores de similaridade incorporados aos planos de execução. Essa característica garante determinismo e características HTAP sem introduzir dependências com o esquema físico, em contraste com *vector stores* de buscas aproximadas.

Arquitetura `CrimeSmartScribe` e Interações Multiagentes. A Figura 2 apresenta a arquitetura da plataforma `CrimeSmartScribe`, suas camadas e as interações entre os componentes. O fluxo inicia-se com a submissão de uma pergunta pelo usuário por meio de uma interface web implementada em Streamlit ① (servindo como camada de apresentação também da resposta final, incluindo conteúdo textual, analítico (*dashboards*) e visual). Ao receber a requisição, o *SessionManager* instancia um objeto `State` que é propagado ao longo de todo o *workflow* e delega o controle da execução ao *AgentManager* ②. Esse componente é responsável por *(i)* materializar o estado da execução em uma instância de `GraphState` interna do framework de agentes *LangChain* e *(ii)* or-

[1]Como os modelos físicos NewSQL variam, os índices disponíveis são diferentes entre si.

questrar os agentes ao longo da execução ③. Ao término do fluxo, o `State` resultante é devolvido ao *SessionManager*, que coordena a construção da resposta final ao usuário. O fluxo de colaborações entrem agentes (grafo direcionado) tem início no nó de entrada ④, responsável por encaminhar a consulta ao agente Supervisor. Esse agente atua como mecanismo de triagem inicial, verificando se a pergunta pode ser respondida com base nos dados internos do sistema. Em seguida, o contexto é enriquecido por uma consulta ao banco vetorial, na qual são recuperadas cinco perguntas semanticamente similares, juntamente com suas respostas previamente validadas. O resultado dessa etapa direciona a primeira condicional do grafo ⑤⑥. Caso os dados na base possam responder à pergunta, o agente *texto-para-SQL* utiliza o contexto de perguntas similares e o esquema baseado em visões para a geração da consulta SQL correspondente à pergunta – Agente *TextToSQL* ⑦. Uma vez validada a sintaxe da tradução, a consulta é encaminhada ao agente *RunQuery*, responsável por executar diretamente a consulta no DW ⑧.

Os resultados são processados pelo agente *Manager* ⑨, que realiza uma inferência para determinar o formato mais apropriado da resposta. Essa decisão conduz à segunda condicional do grafo de interações, a partir da qual podem ser acionados, isolada ou combinadamente, os agentes especializados: *InsightWriter* ⑩, responsável pelas narrativas textuais descritivas; *InsightReasoner* ⑪, encarregado de análises estatísticas e identificação de padrões; e *InsightDrawer* ⑫, dedicado à criação de visualizações acompanhadas de explicações textuais. As saídas produzidas por esses agentes são consolidadas pelo *InsightEditor* ⑬, que atua como camada final de curadoria editorial. Esse agente é responsável por integrar os diferentes artefatos gerados, verificar a consistência entre texto e visualizações, e avaliar o refinamento gráfico. Caso seja identificado ajustes, incluindo divergências de dados e visualização, o fluxo retorna ao *InsightDrawer*. Caso contrário, o *worflow* é concluído e o resultado é repassado para a camada de apresentação.

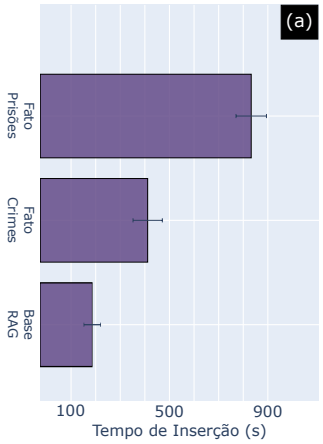
Infraestrutura e *Deploy*. A plataforma foi desenvolvida em Python com arquitetura hexagonal, separando o núcleo do domínio e as camadas externas para garantir a independência em relação a *frameworks* e SGBDs. Para portabilidade foram utilizados princípios de *containerização* que permitiam o *deploy* da plataforma na infraestrutura de nuvem DigitalOcean, por meio de um *droplet* de 2 GB RAM, 25 GB de armazenamento e 01 vCPU. Também como *container*, os *dashboards* são fornecidos via integração com o Metabase v58.4. O NewSQL utilizado foi o CockroachDB serverless com 01 vCPU, 2GB RAM e 10 GiB. Todos os agentes foram configurados com o modelo OpenAI GPT-oss-120b, com limite máximo de 2.5k tokens por geração, top-p = 1 e top-k = 11. Para os agentes *Manager* e *TextToSQL*, a temperatura foi fixada em 0. Para os demais agentes, a temperatura foi definida como 0.5 para favorecer maior diversidade na geração textual.

4. Avaliação Experimental

Esta seção descreve a avaliação experimental do `CrimeSmartScribe`, investigando sua robustez, eficiência e usabilidade frente a diferentes *workloads* de estresse.

Testes de Carga. A infraestrutura HTAP foi submetida a um teste de carga em diferentes cenários de ingestão em lote. A Tabela 1–Figura(a) apresenta os resultados de carga oriunda de um repositório norte-americano para duas tabelas fato representativas (Crimes ≈3.1M, Prisões ≈5M tuplas) e a tabela usada para armazenar dados vetoriais para RAG

Tabela 1. Desempenho para (a) ingestão de dados e (b) consultas OLAP.



Indicador	Fato Crimes	Fato Prisões
Registros Processados	1,65M	3,15M
Tempo de Execução	2,5 s	3,8 s
Dados Lidos	172 MiB	280 MiB
Uso de RAM	39 MiB	44 MiB
Custo de CPU	596 ms	1,0 s
Estratégia / Recursos	∼3.164 RUs	∼5.169 RU

(b) Custo médio de consultas OLAP intensivas.

(≈0.2M tuplas). Como pode ser observado, a inserção tem custo proporcional à cardinalidade para os dados das tabelas fato (vazão entre 3.7 e 3.1k tuplas/s), enquanto que o dado vetorial é mais caro de inserir e indexar (vazão de 492 tuplas/s).

Consultas OLAP. Para avaliar o desempenho do modelo, foram realizadas 100 consultas OLAP baseadas em perguntas em linguagem natural elicitadas com diferentes usuários apresentados ao domínio de dados criminais. O conjunto contempla cenários analíticos não triviais: todas as consultas exigem pelo menos duas junções e um operador multidimensional, enquanto aproximadamente 62% das consultas de referência requerem *Common Table Expressions* e funções de janelas. A Tabela 1(b) apresenta métricas para duas consultas que envolvem as tabelas fato Crime e Prisões – Tabela 2 (Linhas 2 e 4). Para estas, identificou-se a necessidade de índices secundários para otimizar o consumo de recursos (RU), resultando em buscas que processam agregações em até 3,15 milhões de registros consistentemente entre 2,5 e 3,8s, validando a eficiência da indexação.

Visões e RAG – Impacto na Qualidade. A Tabela 2 apresenta alguns exemplos simplificados de consultas e o impacto de cada componente durante o processo de tradução *texto-para-SQL*, indicando um processo incremental para melhoria de qualidade. Para quantificar esse processo, foram definidas 100 consultas indicadas por usuários fora da base RAG, as quais foram submetidas à plataforma em diferentes configurações: *(i)* tradução zero-shot com o LLM e *schema* multidimensional, *(ii)* tradução zero-shot com o LLM com o uso das visões construídas, *(iii)* tradução few-shot com o LLM com o uso das visões construídas, *(ii)* tradução zero-shot com o LLM com o uso de RAG e das visões construídas. A Figura 3(b) apresenta o resultado dessa avaliação incremental agregado, indicando *(i)* O

Tabela 2. Tradução *texto-para-SQL*: Potencial componente a componente

Consulta	*Zero-Shot*	*Zero-Shot* + Visões	*Few-Shot* + Visões	RAG + Visões
Tiroteios por mês e estado em 2018.	Sim	Sim	Sim	Sim
Crime mais comum por faixa etária.	Não	Sim	Sim	Sim
Top3 drogas apreendidas na década 2010.	Não	Sim	Sim	Sim
Top3 crimes por estado nos anos 2000.	Não	Não	Não	Sim

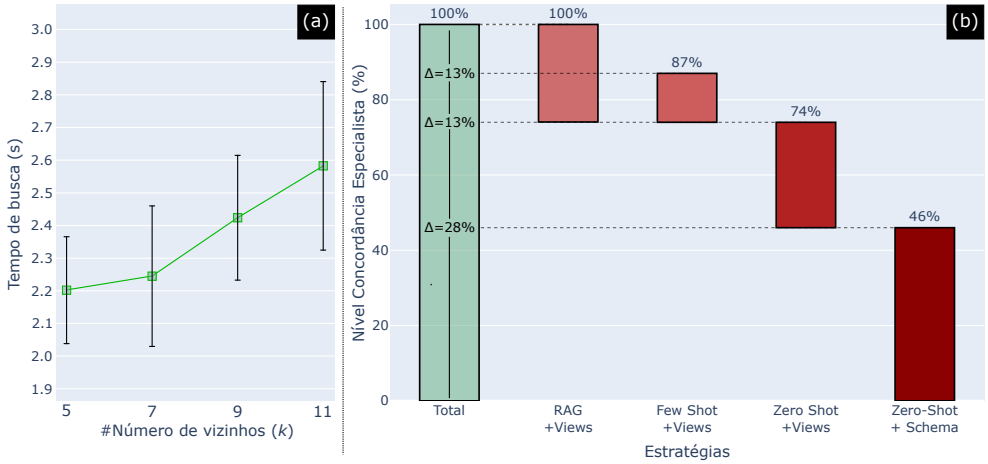


Figura 3. (a) Custo de uma operação RAG em função da quantidade de exemplos (vizinhos – *k*) recuperados. (b) Gráfico cascata destacando o impacto por componente na tradução *texto-para-SQL* feito pelo LLM (média de 100 consultas).

LLM em modo *zero-shot* sobre o *schema* bruto obteve 46% de concordância com especialista, *(ii)* A introdução de Visões simplificadoras permitiu um aumento expressivo de 46% para 74%, *(iii)* O uso de exemplos *few-shot* sobre as visões trouxe um incremento de qualidade, elevando a concordância para 87% – porém, inserindo custos financeiros (quantidade de *tokens*), *(iv)* O acoplamento RAG sobre o *schema* com visões proporcionou outros ganhos adicionais de 13% (com relação ao *few-shot*) e 26% (com relação ao zero-shot), atingindo 100% de concordância – contudo, inserindo custos de desempenho.

Enquanto o *overhead* financeiro introduzido pelo aumento no número de *tokens* no cenário *few-shot* é fixo, a Figura 3(a) mostra que o custo do RAG varia em função do número de vizinhos recuperados. Observa-se um crescimento aproximadamente linear quando se utiliza o índice C-SPANN, indicando que o custo de execução aumenta à medida que a base de perguntas e o número de exemplos recuperados crescem. Embora esse custo tenha permanecido aceitável no experimento conduzido, ele evidencia que o RAG não é uma solução universal para melhorar a qualidade da tradução *texto-para-SQL*. De forma consistente com os resultados da Figura 3(b), os maiores ganhos de concordância decorrem da introdução de visões virtuais, que aumentam significativamente a qualidade sem impacto financeiro ou computacional na execução das consultas. Esses resultados sugerem que, em nível de projeto, visões devem ser priorizadas como mecanismo estrutural de qualidade, com RAG sendo empregado de forma complementar e seletiva, conforme os limites aceitáveis de custo e desempenho. Isso ocorre porque as visões expõem ao LLM um modelo lógico simplificado, enquanto a *engine* de busca opera diretamente sobre o *schema* multidimensional subjacente, preservando eficiência na execução.

Avaliação de Usabilidade. A usabilidade da plataforma foi avaliada por meio do método *System Usability Scale* (SUS), com o objetivo de quantificar a percepção dos usuários quanto à facilidade de uso e à aceitabilidade do sistema. A avaliação foi conduzida em navegadores web Mozilla Firefox 147.0.2, sem a necessidade de instalação de *softwares* adicionais. Participaram do experimento 15 usuários adultos, ensino médio completo, selecionados de forma não probabilística, em linha com avaliações exploratórias de usabilidade adotadas na literatura. O instrumento de avaliação consistiu em uma versão adap-

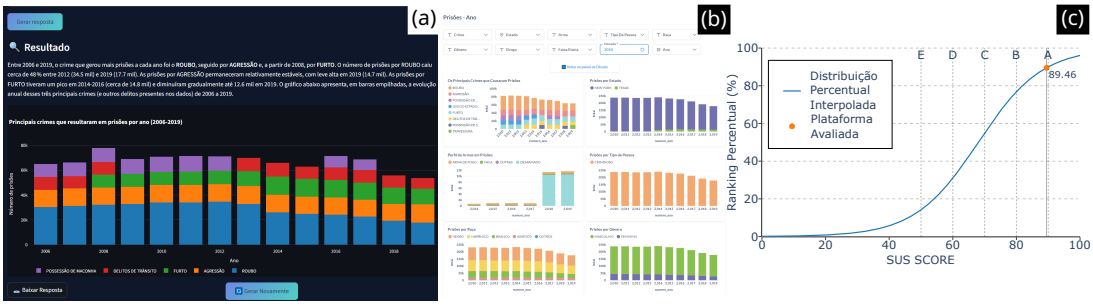


Figura 4. (a) Tela de resposta. (b) *Dashboards*. (c) Avaliação SUS da plataforma.

tada da SUS, composta por dez questões, organizadas em formato *Likert* de cinco pontos, elaboradas com base nos princípios originais do método e preservando sua estrutura conceitual. As sessões de interação foram conduzidas de forma não guiada: após uma breve contextualização sobre o objetivo da plataforma, os participantes foram incentivados a explorá-la livremente, de modo a simular um cenário de uso realista. As respostas foram coletadas por meio de formulário eletrônico. A Figura 4(a-b) mostra exemplo de telas do `CrimeSmartScribe` usadas na avaliação, enquanto a Figura 4(c) apresenta a distribuição percentual aproximada SUS, posicionando a plataforma com uma população de referência. O sistema obteve um *ranking* de 89,46 (Faixa A), associada a alta usabilidade, aceitabilidade e baixo esforço cognitivo durante a interação.

Limitações e Ameaças à Validade. Os resultados indicam o potencial da plataforma, porém também estão sujeitos a limitações e ameaças. Por exemplo, a tradução *texto-para-SQL* foi conduzida em um conjunto finito de consultas de domínio específico, restringindo a generalização para esquemas distintos. Além disso, embora a arquitetura suporte cargas HTAP e vetoriais de forma unificada, os entre *workloads* concorrentes não foram explorados de forma exaustiva devido ao volume de dados disponível. A base RAG foi construída de forma supervisionada, podendo incorrer em viés de cobertura em testes futuros se novas perguntas forem muito semelhantes. Por fim, ressalta-se que a avaliação de usabilidade teve caráter exploratório, com número limitado de participantes, devendo os resultados do SUS serem vistos como indicativos iniciais.

5. Conclusão e Trabalhos Futuros

Este trabalho apresentou o `CrimeSmartScribe`, uma plataforma que integra processamento HTAP e IA generativa para viabilizar o acesso inteligível a dados criminais por meio de consultas em linguagem natural. A avaliação com um conjunto do mundo real mostrou que a proposta apresentou desempenho estável em ingestão e consultas, além de aceitabilidade pelos usuários, destacando a eficácia de respostas narrativas em relação a *dashboards* (também disponíveis). O principal achado experimental refere-se à tradução *texto-para-SQL*: enquanto o uso de LLMs sobre o *schema* bruto resultou em baixa concordância, a introdução de visões promoveu ganhos substanciais, sendo o acoplamento seletivo de técnicas de *few-shot* e RAG responsável por incrementos adicionais nos cenários avaliados, ainda que com custos associados. Isso indica que a abstração lógica do modelo pode constituir o fator estrutural mais relevante para a tradução confiável de consultas, enquanto a contextualização semântica atua como mecanismo complementar de refinamento. Como trabalhos futuros, pretende-se estender a abordagem a bases brasileiras.

Referências

Chen, J., Zhang, L., Xie, Y., Ding, W., Cao, L., Liu, Y., Ding, Y., Li, F., Wu, K., Xiu, H., et al. (2025). Vedb-HTAP: a highly integrated, efficient and adaptive HTAP system. *Proceedings of the VLDB Endowment*, 18(12):4896–4909.

Chien, Y.-C., Wang, K.-D., Wang, W.-Y., and Peng, W.-C. (2025). NEWSAGENT: Benchmarking Multimodal Agents as Journalists with Real-World Newswriting Tasks. *arXiv preprint arXiv:2509.00446*.

Coelho, G. M., Nascimento, E. R., Izquierdo, Y. T., García, G. M., Feijó, L., Lemos, M., Garcia, R. L., de Oliveira, A. R., Pinheiro, J. P., and Casanova, M. A. (2024). Improving the accuracy of text-to-sql tools based on large language models for real-world relational databases. In *International Conference on Database and Expert Systems Applications*, pages 93–107. Springer.

Diakopoulos, N. (2019). *Automating the news: How algorithms are rewriting the media*. Harvard University Press.

Dong, Y., Jiang, X., Jin, Z., and Li, G. (2024). Self-collaboration code generation via ChatGPT. *ACM Transactions on Software Engineering and Methodology*, 33(7):1–38.

Fan, W., Ding, Y., Ning, L., Wang, S., Li, H., Yin, D., Chua, T.-S., and Li, Q. (2024). A survey on rag meeting llms: Towards retrieval-augmented large language models. In *Proceedings of the ACM Conference on Knowledge Discovery and Data Mining*, pages 6491–6501.

Ferber, J. and Weiss, G. (1999). *Multi-agent systems: an introduction to distributed artificial intelligence*, volume 1. Addison-wesley Reading.

Fürst, J., Kosten, C., Nooralahzadeh, F., Zhang, Y., and Stockinger, K. (2024). Evaluating the data model robustness of text-to-sql systems based on real user queries. *arXiv preprint arXiv:2402.08349*.

García-Gil, D., Ramírez-Gallego, S., García, S., and Herrera, F. (2017). A comparison on scalability for batch big data processing on Apache Spark and Apache Flink. *Big Data Analytics*, 2(1):1.

Greenspan, R. L., Baggett, L., and B. Boutwell, B. (2024). Open science practices in criminology and criminal justice journals. *Journal of Experimental Criminology*, pages 1–21.

Hidalgo, J. J. P. and Centeno, J. A. S. (2023). Environmental scanning of cocaine trafficking in Brazil: Evidence from geospatial intelligence and natural language processing methods. *Science & Justice*, 63(6):689–723.

Jamil, H. M. and Rubaiat, S. Y. (2024). Online Digital Investigative Journalism Using SociaLens. In *International Conference on Information Integration and Web Intelligence*, pages 103–117. Springer.

Luo, Y., Li, G., Fan, J., Chai, C., and Tang, N. (2025). Natural language to sql: State of the art and open problems. *Proceedings of the VLDB Endowment*, 18(12):5466–5471.

Maltezos, V., Kyrychenko, R., and Knuutila, A. (2025). How can AI agents support journalists' work? An experiment with designing an LLM-driven intelligent reporting system. *arXiv preprint arXiv:2510.01193*.

Miehling, E., Ramamurthy, K. N., Varshney, K. R., Riemer, M., Bouneffouf, D., Richards, J. T., Dhurandhar, A., Daly, E. M., Hind, M., Sattigeri, P., et al. (2025). Agentic AI needs a systems theory. *arXiv preprint arXiv:2503.00237*.

Mohan, R. K., Kanmani, R. R. S., Ganesan, K. A., and Ramasubramanian, N. (2024). Evaluating NoSQL Databases for OLAP Workloads: A Benchmarking Study of MongoDB, Redis, Kudu and ArangoDB. *arXiv preprint arXiv:2405.17731*.

Mtchedlidze, J. (2024). Technical Expertise in Newsrooms: Understanding Data Journalists' Roles and Practices. *Journalism and Media*, 5(3):1316–1328.

Pan, J. J., Wang, J., and Li, G. (2024). Survey of vector database management systems. *The VLDB Journal*, 33(5):1591–1615.

Shi, L., Tang, Z., Zhang, N., Zhang, X., and Yang, Z. (2025). A survey on employing large language models for text-to-sql tasks. *ACM Computing Surveys*, 58(2):1–37.

Silva, R., Chagnoux, M., and Vassiliadis, P. (2022). Data Narrative Crafting via a Comprehensive and Well-Founded Process. In *Advances in Databases and Information Systems*, page 347. Springer.

Siqueira, P., Dias, R., Silva-Leite, J., Mann, P., Salvador, R., de Oliveira, D., and Bedo, M. (2025). Warehousing Data for Brand Health and Reputation with AI-Driven Scores in NewSQL Architectures: Opportunities and Challenges. *International Conference on Enterprise Information Systems*, pages 1–12.

Taft, R., Sharif, I., Matei, A., VanBenschoten, N., Lewis, J., Grieger, T., Niemi, K., Woods, A., Birzin, A., Poss, R., et al. (2020). Cockroachdb: The resilient geo-distributed sql database. In *Proceedings of the 2020 ACM SIGMOD international conference on management of data*, pages 1493–1509.

Zaharia, M. (2025). Bringing the Operational and Analytical Worlds Together with Lakebase. *Proceedings of the VLDB Endowment*, 18(12):5539–5539.