UNIVERSIDADE FEDERAL DE MINAS GERAIS
ESCOLA DE ENFERMAGEM
PROGRAMA DE PÓS-GRADUAÇÃO EM
GESTÃO DE SERVIÇOS DE SAÚDE
MESTRADO PROFISSIONAL EM GESTÃO DE SERVIÇOS DE SAÚDE

Fabíola de Oliveira Simões

PRODUTO TÉCNICO/TECNOLÓGICO RESULTANTE DO TRABALHO “Aplicação de Processamento de Linguagem Natural como ferramenta para suporte à Gestão Farmacêutica em ambiente hospitalar”

Belo Horizonte
2026

Fabíola de Oliveira Simões

PRODUTO TÉCNICO/TECNOLÓGICO RESULTANTE DO TRABALHO “Aplicação de Processamento de Linguagem Natural como ferramenta para suporte à Gestão Farmacêutica em ambiente hospitalar”

Produto técnico/tecnológico resultante do Trabalho de Conclusão de Mestrado apresentado ao curso de Mestrado Profissional do Programa de Pós-Graduação em Gestão de Serviços de Saúde, da Escola de Enfermagem da Universidade Federal de Minas Gerais, como requisito parcial para obtenção do grau de Mestre em Gestão de Serviços de Saúde.
Área de concentração: Gestão de Serviços de Saúde.

Linha de pesquisa: Tecnologias Gerenciais em Saúde

Orientador: Prof. Dr. Raphael Augusto Teixeira de Aguiar

Coorientador: Prof. Dr. Leonardo Chaves Dutra da Rocha

Belo Horizonte
2026

RESUMO

A gestão de farmácias hospitalares envolve volumes significativos de dados e requer acesso rápido a informações confiáveis para apoiar os processos de tomada de decisão. Este estudo metodológico teve como objetivo explorar a aplicação do Processamento de Linguagem Natural (PLN) na gestão de farmácias hospitalares por meio do desenvolvimento de um protótipo de *chatbot* baseado em um Modelo Amplo de Linguagem (LLM) da família Qwen. A pesquisa foi conduzida em duas fases complementares: a primeira fase de caráter exploratório, dedicada à identificação das principais solicitações de informação relacionadas à gestão de estoques e uma segunda fase de desenvolvimento e avaliação funcional do protótipo, implementado em Python. A avaliação da ferramenta focou em um conjunto estruturado de perguntas representativas das necessidades comuns dos profissionais, resultantes da análise das respostas dos participantes durante a primeira fase. Os resultados mostraram uma taxa de sucesso geral de execução de 85%, indicando que o modelo foi capaz de gerar consultas SQL válidas na maioria dos casos. O desempenho mostrou-se satisfatório para perguntas simples de baixa a moderada complexidade, que produziram respostas consistentes e acionáveis. No entanto, uma proporção significativa de consultas mais complexas não levou a respostas semanticamente corretas, apesar da execução técnica bem-sucedida, evidenciando principalmente limitações na interpretação de regras de negócio implícitas. De modo geral, o protótipo demonstrou viabilidade técnica e potencial para aplicação prática, ao mesmo tempo que ressaltou a necessidade de ajustes metodológicos e refinamentos do modelo para melhorar a precisão das respostas em cenários de tomada de decisão mais complexos.

Palavras-chave: Processamento de Linguagem Natural; Sistemas Inteligentes; Serviço de Farmácia Hospitalar; Inteligência Artificial.

ABSTRACT

Hospital pharmacy management involves significant volumes of data and requires rapid access to reliable information to support decision-making processes. This methodological study aimed to explore the application of Natural Language Processing (NLP) in hospital pharmacy management through the development of a chatbot prototype based on a Large Language Model (LLM) from the Qwen family. The research was conducted in two complementary phases: the first, exploratory phase, dedicated to identifying the main information requests related to inventory management, and a second phase of development and functional evaluation of the prototype, implemented in Python. The evaluation of the tool focused on a structured set of questions representative of the common needs of professionals, resulting from the analysis of participants' responses during the first phase. The results showed an overall execution success rate of 85%, indicating that the model was able to generate valid SQL queries in most cases. Performance proved satisfactory for simple questions of low to moderate complexity, which produced consistent and actionable responses. However, a significant proportion of more complex queries did not lead to semantically correct answers, despite successful technical execution, mainly highlighting limitations in the interpretation of implicit business rules. Overall, the prototype demonstrated technical feasibility and potential for practical application, while also emphasizing the need for methodological adjustments and model refinements to improve the accuracy of responses in more complex decision-making scenarios.

Keywords: Natural Language Processing; Intelligent Systems; Pharmacy Service, Hospital; Artificial Intelligence.

LISTA DE ILUSTRAÇÕES

Figura 1 – Fluxo simplificado 11

Figura 2 – Esquema de funcionamento do sistema RAG 12

Figura 3 – Como o sistema RAG funciona 13

Quadro 1 – Síntese das métricas de avaliação 17

LISTA DE ABREVIATURAS E SIGLAS

LLM	Large Language Models (Modelos Amplos de Linguagem)
RAG	Recuperação Aumentada por Geração
SQL	Structured Query Language (Linguagem de Consulta Estruturada)
TI	Tecnologia da Informação

SUMÁRIO

1 INTRODUÇÃO 7
2 OBJETIVO DO PRODUTO TÉCNICO 8
3 CARACTERIZAÇÃO DO PROTÓTIPO DESENVOLVIDO.....9
3.1 Arquitetura e funcionamento do sistema 9
3.2 Conversão de linguagem natural em SQL.....10
3.3 Sistema de Recuperação Aumentada por Geração (RAG) 11
3.3.1 Modelo de inteligência artificial utilizado.....13
3.3.2 Parâmetros configuráveis do sistema.....14
3.3.3 Estrutura do banco de dados 14
3.3.4 Vantagens da abordagem proposta 15
4 MÉTRICAS DE AVALIAÇÃO DO MODELO.....17
5 DISPONIBILIZAÇÃO E REPRODUTIBILIDADE DO PROTÓTIPO.....20
6 CONSIDERAÇÕES FINAIS 21
REFERÊNCIAS.....22

RELATÓRIO TÉCNICO

PROTÓTIPO DE CONVERSÃO DE LINGUAGEM NATURAL EM SQL COM USO DE RECUPERAÇÃO AUMENTADA POR GERAÇÃO (RAG)

1 INTRODUÇÃO

A crescente digitalização dos serviços de saúde tem ampliado de forma significativa o volume de dados armazenados em sistemas de informação hospitalares (Yang *et al*., 2025). Embora esses dados sejam fundamentais para o apoio à gestão, à tomada de decisão e ao planejamento assistencial, seu acesso permanece, em grande parte, restrito a profissionais com conhecimento técnico específico em linguagens de consulta a bancos de dados, como o SQL (Linguagem de Consulta Estruturada). Esse cenário impõe barreiras à utilização plena das informações disponíveis, especialmente por gestores e profissionais da área da saúde que não possuem formação em tecnologia da informação.

No contexto da gestão farmacêutica hospitalar, essa limitação torna-se ainda mais sensível, uma vez que decisões relacionadas a estoque, consumo, cobertura de medicamentos e materiais impactam diretamente a segurança do paciente, a eficiência operacional e os custos institucionais (Raimundo; Dias; Guerra, 2015; Lima; Câmara, 2016; Bruns; Luiza; Oliveira, 2014). Diante desse desafio, tecnologias baseadas em Processamento de Linguagem Natural e modelos de linguagem de grande porte surgem como alternativas para aproximar usuários não técnicos dos sistemas de dados estruturados.

Este relatório técnico apresenta o protótipo Text2SQL desenvolvido como produto técnico associado a dissertação de mestrado profissional intitulada “Aplicação de Processamento de Linguagem Natural como ferramenta para suporte à Gestão Farmacêutica em ambiente hospitalar”, alinhado às diretrizes de produção técnica aplicadas a cursos stricto sensu de natureza profissional (CAPES, 2019). O protótipo tem como finalidade permitir que perguntas formuladas em linguagem natural sejam automaticamente convertidas em consultas SQL executáveis, com apoio de uma arquitetura baseada em Recuperação Aumentada por Geração (Retrieval-Augmented Generation – RAG).

2 OBJETIVO DO PRODUTO TÉCNICO

O objetivo geral deste produto técnico é apresentar e documentar o desenvolvimento de um protótipo de sistema capaz de converter perguntas em linguagem natural para consultas SQL, utilizando Modelo Amplo de Linguagem (LLM) apoiado por um mecanismo de recuperação de contexto estruturado.

Como objetivos específicos, destacam-se:

- Descrever a arquitetura técnica do protótipo Text2SQL;
- Explicar o funcionamento do sistema de Recuperação Aumentada por Geração aplicado ao contexto de bancos de dados hospitalares;
- Apresentar as principais funcionalidades implementadas no protótipo;
- Documentar as métricas utilizadas para avaliação do desempenho do modelo, em nível conceitual;
- Disponibilizar informações que favoreçam a reprodutibilidade e a evolução futura da solução proposta.

3 CARACTERIZAÇÃO DO PROTÓTIPO DESENVOLVIDO

O protótipo Text2SQL consiste em uma aplicação computacional desenvolvida em linguagem Python, projetada para permitir a interação entre usuários e bases de dados relacionais por meio de perguntas formuladas em linguagem natural (Silva; Silva; Silva, 2025). O sistema foi concebido para operar sobre bancos de dados PostgreSQL e utilizou o LLM Qwen3-32B-AWQ, por sua natureza *Open Source*, capacidade de compreensão semântica em língua portuguesa e desempenho satisfatório em tarefas de geração de texto estruturado.

O público-alvo do protótipo compreende gestores e profissionais da área da saúde que necessitam acessar informações armazenadas em bancos de dados hospitalares, mas que não possuem domínio técnico da linguagem SQL. O protótipo foi desenvolvido com foco no domínio da gestão farmacêutica hospitalar, considerando estruturas de dados relacionadas a estoque, cadastro de materiais e medicamentos, movimentações e indicadores de cobertura.

Do ponto de vista funcional, o sistema permite a geração automática de consultas SQL, a execução dessas consultas no banco de dados, a recuperação dos resultados e a avaliação da qualidade das respostas geradas, além de disponibilizar uma interface web para visualização e análise dos resultados.

3.1 Arquitetura e funcionamento do sistema

O protótipo Text2SQL foi desenvolvido a partir de uma organização que distribui suas funcionalidades em partes bem definidas, cada uma responsável por uma etapa do processo de transformação das perguntas em informações úteis para o usuário. Essa forma de organização permite que o sistema funcione de maneira integrada e facilita sua manutenção e adaptação a diferentes realidades institucionais.

De modo geral, o sistema recebe perguntas formuladas em linguagem natural (texto em português), identifica quais informações do banco de dados são necessárias para respondê-las e realiza automaticamente a consulta correspondente. Todo esse processo ocorre de forma transparente para o usuário, que não precisa conhecer a estrutura do banco de dados nem dominar a linguagem SQL para obter os resultados desejados.

O acesso ao protótipo pode ocorrer tanto por meio de uma interface de linha de comando quanto por uma interface web (Shiny). A primeira é voltada principalmente

ao uso técnico e à execução de testes, enquanto a interface web foi pensada para facilitar a interação de usuários não especialistas. Independentemente da forma de acesso, o funcionamento interno do sistema permanece o mesmo, assegurando que as respostas geradas sejam consistentes.

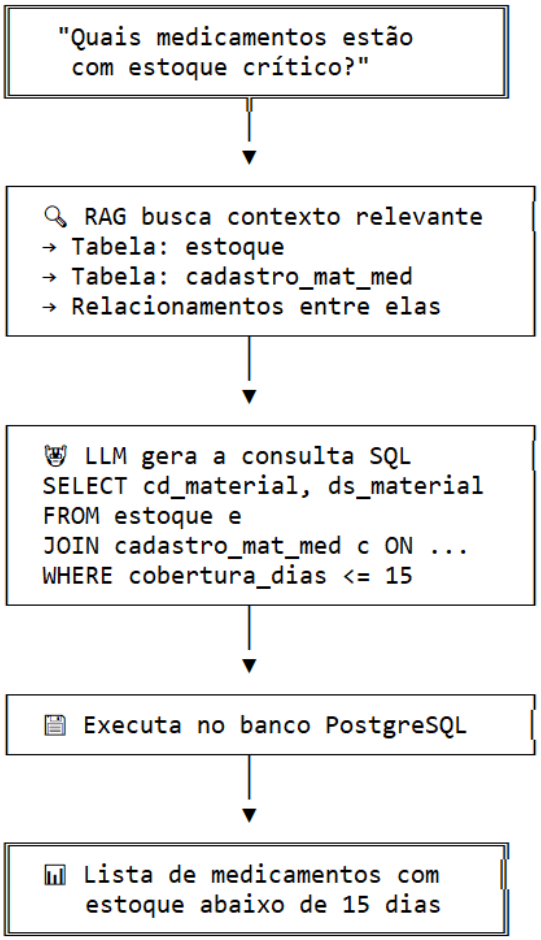
Após o envio da pergunta, o sistema realiza o processamento necessário, executa a consulta no banco de dados e apresenta os resultados de forma organizada. Dessa maneira, o protótipo atua como um intermediário entre o usuário e o banco de dados, simplificando o acesso à informação e ampliando o uso dos dados disponíveis para apoio à gestão.

3.2 Conversão de linguagem natural em SQL

O núcleo do protótipo é o mecanismo de conversão de linguagem natural em SQL, realizado por meio de um LLM. Esse modelo recebe como entrada a pergunta formulada pelo usuário em português e é instruído a produzir uma consulta SQL correspondente, respeitando a estrutura do banco de dados e regras de negócio previamente definidas.

Entretanto, a utilização direta de modelos de linguagem para geração de SQL apresenta limitações importantes, especialmente quando o modelo não dispõe de informações suficientes sobre o esquema do banco de dados, como nomes de tabelas, colunas e relacionamentos. Para mitigar essas limitações, o protótipo incorpora um mecanismo de Recuperação Aumentada por Geração (RAG), abordagem que tem se mostrado eficaz para aumentar a confiabilidade e a rastreabilidade das respostas geradas por modelos de linguagem (Soares, 2024; Yang *et al*., 2025). A Figura 1 esquematiza o processo de construção da ferramenta:

Figura 1 – Fluxo simplificado

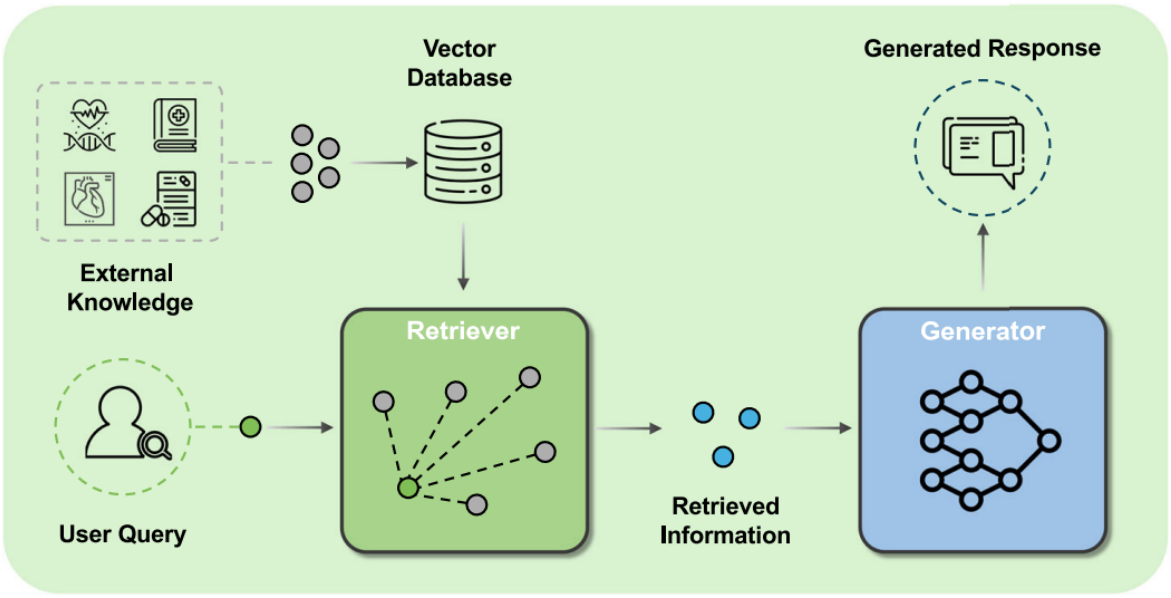


Fonte: Elaborado pela equipe de TI responsável pelo desenvolvimento do protótipo.

3.3 Sistema de Recuperação Aumentada por Geração (RAG)

A técnica de Recuperação Aumentada por Geração é responsável por fornecer ao modelo de linguagem informações contextuais relevantes sobre o banco de dados antes da geração da consulta SQL, seguindo a arquitetura clássica de indexação, recuperação e geração descrita na literatura (Yang *et al.*, 2025). A Figura 2 ilustra o funcionamento geral dessa abordagem. No modelo apresentado, a pergunta do usuário é inicialmente utilizada para recuperar informações relevantes a partir de uma base de conhecimento externa. Em seguida, esse conjunto de informações é incorporado ao processo de geração da resposta pelo LLM.

Figura 2 – Esquema de funcionamento do sistema RAG



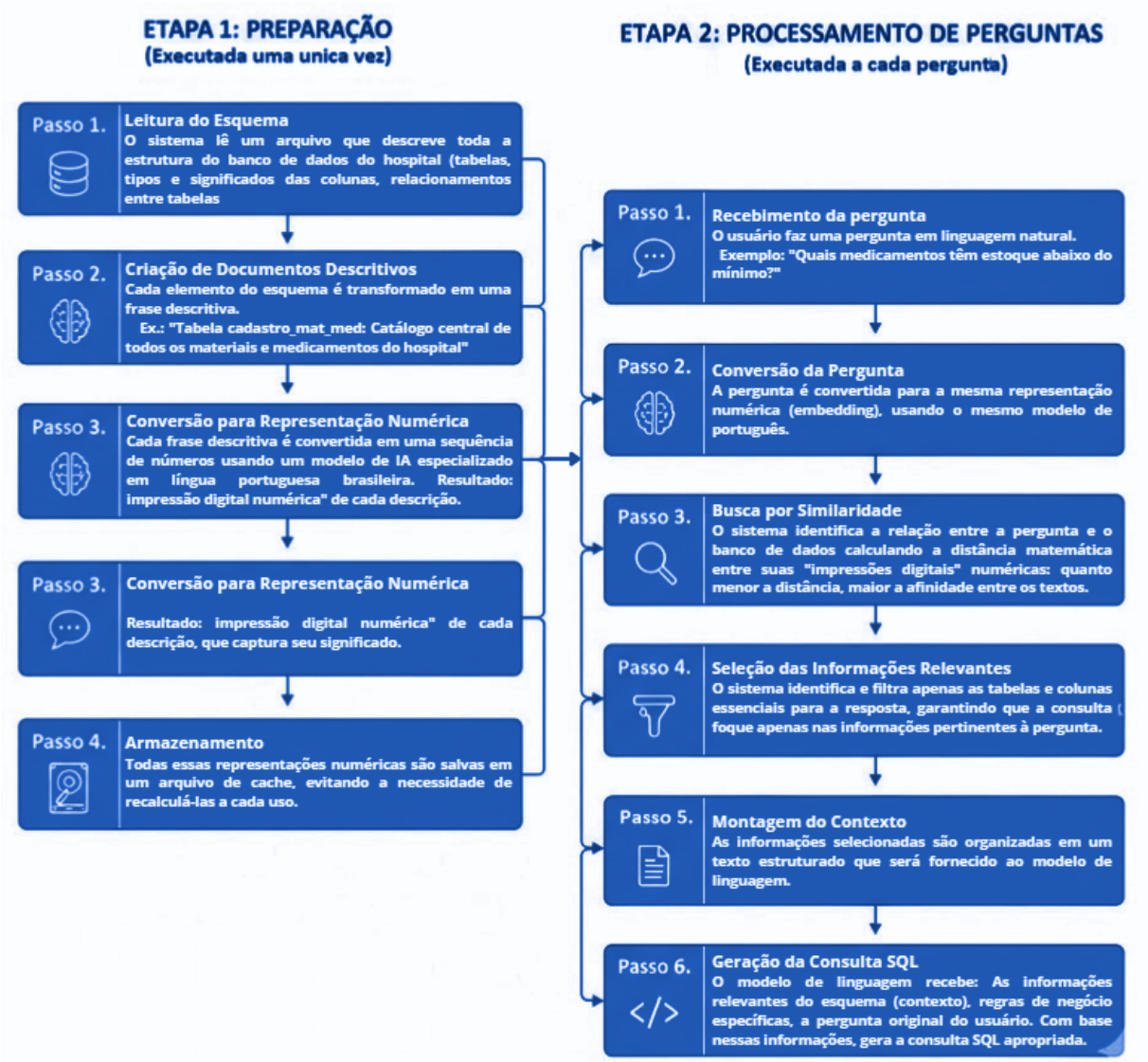
Fonte: Yang *et al*. (2025).

Na fase de indexação, essas informações são transformadas em representações vetoriais por meio de modelos de *embeddings* treinados para o idioma português. Os vetores gerados são armazenados em um índice vetorial, permitindo a recuperação eficiente de informações com base em similaridade semântica. *Embeddings* são representações numéricas de textos. Imagine que cada palavra ou frase é convertida em uma "coordenada" em um espaço matemático multidimensional. Palavras com significados semelhantes ficam "próximas" nesse espaço. Por exemplo: as palavras "medicamento", "remédio" e "fármaco" teriam coordenadas muito próximas entre si, enquanto a palavra "cadeira" ficaria distante. Isso permite que o computador entenda que textos são semanticamente similares, mesmo usando palavras diferentes. Quando o usuário submete uma pergunta em linguagem natural, o sistema gera um vetor correspondente à pergunta e realiza uma busca por similaridade no índice vetorial.

Como resultado, são recuperados apenas os elementos do esquema do banco de dados mais relevantes para aquela pergunta específica. Esse conjunto de informações é então incorporado ao *prompt* fornecido ao modelo de linguagem, aumentando a probabilidade de geração de uma consulta SQL correta e coerente. A Figura 3 esquematiza este fluxo no protótipo desenvolvido.

Essa abordagem permite reduzir ambiguidades, aumentar a confiabilidade das consultas geradas e melhorar a transparência do processo de geração, uma vez que o contexto utilizado pelo modelo pode ser explicitamente identificado.

Figura 3 – Como o sistema RAG funciona



Fonte: Elaborado pela autora.

3.3.1 Modelo de inteligência artificial utilizado

Embora o modelo de linguagem responsável pela geração das consultas SQL seja o Qwen3-32B-AWQ, o sistema utiliza um segundo modelo, o BERTimbau (neuralmind/bert-large-portuguese-cased),[1] para a etapa de recuperação de contexto. Esse modelo é empregado para converter textos em representações numéricas,

[1] https://huggingface.co/neuralmind/bert-large-portuguese-cased

permitindo comparar semanticamente a pergunta do usuário com descrições do esquema do banco de dados. Dessa forma, o BERTimbau atua exclusivamente na seleção das informações relevantes, enquanto o Qwen é responsável pela geração final da consulta SQL.

O modelo BERTimbau foi escolhido por:

- Especialização em português: Foi treinado especificamente com textos em português brasileiro, compreendendo melhor expressões, gírias e estruturas gramaticais do nosso idioma.
- Tamanho e capacidade: Possui 335 milhões de parâmetros ajustáveis, o que lhe confere alta capacidade de compreensão textual.
- Sensibilidade a maiúsculas: Diferencia “Medicamento” de “medicamento”, preservando informações importantes presentes em nomes técnicos.
- Treinamento abrangente: Foi treinado em um grande volume de textos brasileiros, incluindo artigos, notícias e documentos técnicos.

3.3.2 Parâmetros configuráveis do sistema

O sistema permite ajustar diversos parâmetros para otimizar seu funcionamento:

- Limiar de similaridade (similarity_threshold): Define quão “parecido” um resultado precisa ser para ser considerado relevante. Valor padrão: 0.3 (em uma escala de 0 a 1);
- Número máximo de tabelas (max_tables): Quantas tabelas podem ser incluídas no contexto. Valor padrão: 5 tabelas;
- Colunas por tabela (max_columns_per_table): Quantas colunas de cada tabela são incluídas. Valor padrão: 10 colunas;
- Tamanho máximo do contexto (max_context_length): Limite de caracteres do contexto gerado. Valor padrão: 32.768 caracteres.

3.3.3 Estrutura do banco de dados

O esquema do banco de dados foi documentado em um formato estruturado (YAML) contendo, para cada tabela:

- Descrição da tabela: Explicação do propósito e conteúdo da tabela
  Exemplo: “Catálogo central de todos os materiais/medicamentos do hospital”
- Lista de colunas: Cada coluna com seu tipo de dado e descrição
  Exemplo: “cd_material (número inteiro) – Código único do item”

- Chave primária: Identificador único de cada registro na tabela
- Chaves estrangeiras: Relacionamentos com outras tabelas

Exemplo: “A tabela de estoque se relaciona com o cadastro de materiais através do código do material”

- Índices: Colunas otimizadas para buscas rápidas

Esta documentação detalhada permite que o sistema compreenda não apenas a estrutura técnica, mas também o significado de cada elemento no contexto hospitalar.

3.3.4 Vantagens da abordagem proposta

A adoção de uma arquitetura baseada em RAG apresenta um conjunto de vantagens relevantes no contexto de sistemas que integram modelos de linguagem e bases de dados estruturadas. Essas vantagens dizem respeito tanto ao desempenho computacional quanto à qualidade das respostas geradas e à possibilidade de aplicação do sistema em diferentes contextos institucionais.

a) Economia de recursos computacionais

A abordagem proposta evita o envio integral da documentação do banco de dados ao modelo de linguagem a cada solicitação. Em vez disso, apenas as informações consideradas relevantes para a pergunta formulada pelo usuário são selecionadas e fornecidas como contexto. Essa estratégia contribui para a redução do volume de dados processados, resultando em menor tempo de resposta e em uso mais eficiente dos recursos computacionais, aspecto especialmente importante em ambientes que demandam escalabilidade e controle de custos.

b) Aumento da precisão das respostas

Ao receber um conjunto de informações direcionado e alinhado à pergunta realizada, o modelo de linguagem opera com menor risco de ambiguidades. A restrição do contexto às tabelas, colunas e relacionamentos pertinentes reduz a probabilidade de geração de consultas incorretas ou inconsistentes, favorecendo respostas mais precisas e compatíveis com a estrutura real do banco de dados.

c) Escalabilidade do sistema

A arquitetura adotada permite que o sistema mantenha desempenho satisfatório mesmo em bancos de dados de grande porte, com elevado número de

tabelas e atributos. A busca por similaridade semântica, utilizada para a recuperação do contexto relevante, apresenta custo computacional reduzido quando comparada à análise exaustiva de todo o esquema do banco, tornando a solução adequada para cenários complexos e em expansão.

d) Compreensão semântica das perguntas

Diferentemente de abordagens baseadas exclusivamente em correspondência literal de termos, o sistema considera o significado das palavras utilizadas na pergunta. Dessa forma, expressões semanticamente relacionadas são reconhecidas como equivalentes, mesmo quando não há coincidência exata de vocabulário. Isso permite, por exemplo, que perguntas sobre “remédios vencidos” sejam associadas a informações relacionadas a medicamentos e prazos de validade, ampliando a capacidade de interpretação do sistema.

e) Adaptabilidade a diferentes contextos

A solução proposta apresenta elevada flexibilidade, uma vez que sua adaptação a novos bancos de dados depende, essencialmente, da atualização do arquivo de esquema que descreve a estrutura dos dados. Essa característica dispensa alterações no código-fonte principal do sistema, facilitando sua reutilização em diferentes instituições e contextos de aplicação, o que é particularmente relevante em ambientes organizacionais heterogêneos.

4 MÉTRICAS DE AVALIAÇÃO DO MODELO

Como etapa fundamental do desenvolvimento do protótipo, foi construída uma base de referência denominada *ground truth*, composta por consultas SQL elaboradas e validadas pela equipe de Tecnologia da Informação (TI). O *ground truth* corresponde a um conjunto de consultas consideradas corretas e representativas das perguntas de interesse, servindo como padrão para comparação e avaliação das consultas geradas automaticamente pelo modelo.

A avaliação de modelos em Machine Learning é um aspecto crítico que não pode ser negligenciado. Para garantir que esses modelos sejam capazes de explicar e prever fenômenos com precisão, é muito importante utilizar as métricas corretas. O desempenho de cada pergunta do conjunto de avaliação foi analisado por meio de métricas quantitativas para avaliação de sistemas de recuperação de informação e geração automática de consultas, tais como **acurácia, precisão**, ***recall*** e ***F1-score*** (Quadro 1). Essas métricas foram calculadas a partir da comparação entre as consultas SQL geradas automaticamente pelo modelo e a base de referência (*ground truth*) previamente validada.

Quadro 1 – Síntese das métricas de avaliação

Métrica	Descrição	Fórmula
Acurácia	Indica uma performance geral do modelo. Dentre todas as classificações, quantas o modelo classificou corretamente.	$\frac{VP + VN}{VP + FV + FP + FN}$
Precisão	Dentre todas as classificações da classe positiva que o modelo fez, quantas estão corretas.	$\frac{VP}{VP + FP}$
Recall /Revocação/ Sensibilidade:	Dentre todas as classificações da classe positiva como valor esperado, quantas estão corretas.	$\frac{VP}{VP + FN}$
F1-Score	Média harmônica entre precisão e *Recall*.	$\frac{2 * Precisão * Recall}{Precisão + Recall}$

Fonte: Pinho; Gaspar; Sassi (2024).

a) Acurácia

A acurácia representa a proporção geral de acertos do sistema, considerando tanto os itens corretamente retornados quanto os itens incorretamente incluídos, devendo ser interpretada com cautela em tarefas de recuperação de informação. A fórmula da acurácia é a proporção de previsões corretas em relação ao total de

previsões realizadas. Para entender os acertos, utiliza-se a matriz de confusão,[2] que separa Verdadeiros Positivos (VP), Verdadeiros Negativos (VN), Falsos Positivos (FP) e Falsos Negativos (FN). Embora seja amplamente utilizada, sua interpretação deve ser realizada com cautela em problemas de recuperação de informação, sobretudo quando há desbalanceamento entre respostas corretas e incorretas (ex: 99% spam, 1% não-spam), o modelo pode ter alta acurácia (99%) sem realmente aprender a identificar a classe minoritária) (Pinho; Gaspar; Sassi, 2024; Ferri; Hernández-Orallo; Modroiu, 2009).

b) Precisão

A precisão é uma métrica que indica a proporção de itens retornados pelo sistema que estão corretos em relação ao total de itens apresentados como resposta, sendo amplamente utilizada na avaliação de sistemas de recuperação de informação e aprendizado de máquina. A fórmula da precisão é o número de verdadeiros positivos dividido pela soma dos verdadeiros positivos e falsos positivos. É particularmente importante quando uma previsão positiva seria dispendiosa se se revelasse errada (Pinho; Gaspar; Sassi, 2024; Ferri; Hernández-Orallo; Modroiu, 2009).

No contexto de perguntas que resultam em listas de registros, a precisão permite avaliar o grau de exatidão das respostas fornecidas pelo modelo.

c) *Recall*

O *recall* mede a proporção de itens corretos que foram efetivamente recuperados pelo sistema em relação ao total de itens esperados, refletindo a capacidade do modelo em não omitir respostas relevantes. Essa métrica é particularmente relevante em contextos nos quais a omissão de informações pode comprometer a tomada de decisão (Pinho; Gaspar; Sassi, 2024; Ferri; Hernández-Orallo; Modroiu, 2009).

---

[2] **Verdadeiro Positivo** corresponde às situações em que o valor esperado é positivo e o modelo identifica corretamente esse caso, retornando um resultado também positivo, o que caracteriza um acerto da predição realizada.

Verdadeiro Negativo refere-se aos casos em que o resultado esperado é negativo e o modelo corretamente retorna um valor negativo, caracterizando uma predição correta.

Falso Positivo ocorre quando o modelo indica um resultado positivo para um caso cujo valor real é negativo, caracterizando uma classificação incorreta.

Falso Negativo ocorre quando o modelo classifica um caso como negativo, apesar de o valor real ser positivo, caracterizando uma predição incorreta (Pinho; Gaspar; Sassi, 2024).

A fórmula para *recall* é o número de verdadeiros positivos dividido pela soma de verdadeiros positivos e falsos negativos. Uma alta pontuação de *recall* mostra que o modelo está capturando a maioria dos casos positivos reais nos dados.

d) *F1-score*

O *F1-score* corresponde à média harmônica entre precisão e *recall*, sendo obtido pelo dobro do produto dessas métricas dividido pela soma de ambas. Essa medida é particularmente relevante em contextos nos quais tanto os falsos positivos quanto os falsos negativos possuem impacto significativo, oferecendo uma avaliação mais equilibrada do desempenho do modelo, especialmente em conjuntos de dados desbalanceados (Pinho; Gaspar; Sassi, 2024; Ferri; Hernández-Orallo; Modroiu, 2009).

5 DISPONIBILIZAÇÃO E REPRODUTIBILIDADE DO PROTÓTIPO

Com o objetivo de promover a transparência, a reprodutibilidade científica e a possibilidade de evolução contínua da solução proposta, o código-fonte do protótipo Text2SQL foi disponibilizado em um repositório GitHub, prática recomendada para produtos técnicos e pesquisas aplicadas em ciência de dados e saúde digital. O repositório contém a estrutura completa do projeto, incluindo os módulos de geração de consultas, o sistema de recuperação de contexto, os scripts de avaliação e a interface web.

O acesso ao repositório permite que outros pesquisadores e profissionais da área da saúde compreendam a implementação do protótipo, adaptem-no a diferentes contextos institucionais e contribuam para seu aprimoramento futuro. Adicionalmente, eventuais melhorias de desempenho ou ajustes no protótipo serão registradas no GitHub, permitindo o acompanhamento da evolução do projeto. É importante destacar que os resultados apresentados refletem o estágio atual do desenvolvimento, podendo não ser definitivos, já que fatores como o tempo disponível para execução e a conclusão para apresentação à banca final podem influenciar o desempenho observado.

O Link para acesso ao repositório GitHub*:* https://github.com/fabiolaoliveira-2026/Projeto-Prot-tipo-Gest-o-Farmac-utica.git

6 CONSIDERAÇÕES FINAIS

Este relatório técnico apresentou o protótipo Text2SQL desenvolvido como produto técnico de um trabalho de conclusão de mestrado, com foco na conversão de linguagem natural em consultas SQL apoiada por um mecanismo de Recuperação Aumentada por Geração. O documento descreveu a arquitetura do sistema, seu funcionamento, os principais componentes técnicos e as métricas utilizadas para avaliação do desempenho do modelo.

O protótipo proposto demonstra o potencial de aplicação de LLMs e técnicas de RAG no apoio à gestão farmacêutica hospitalar, contribuindo para a ampliação do acesso a informações estratégicas por usuários não técnicos. Como produto técnico, a solução apresentada constitui uma base para futuras evoluções, adaptações e estudos aplicados no contexto dos sistemas de informação em saúde.

REFERÊNCIAS

BRUNS, Suelma de Fátima; LUIZA, Vera Lucia; OLIVEIRA, Egléubia Andrade de. Gestão da assistência farmacêutica em municípios do estado da Paraíba (PB): olhando a aplicação de recursos públicos. *Revista de Administração Pública*, v. 48, n. 3, p. 745–765, 2014. Doi: https://doi.org/10.1590/0034-76121502. Acesso em: 3 jan. 2026.

COORDENAÇÃO DE APERFEIÇOAMENTO DE PESSOAL DE NÍVEL SUPERIOR (CAPES). *Produção técnica*: Grupo de Trabalho. Brasília: CAPES, 2019. Disponível em: https://www.gov.br/capes/pt-br/centrais-de-conteudo/10062019-producao-tecnica-pdf/view. Acesso em: 18 jan. 2026.

FERRI, C.; HEERNÁNDEZ-ORALLO, J.; MODROIU, R. An Experimental Comparison of Performance Measures for Classification. *Pattern Recognition Letters*, v. 30, n. 1, p. 27–38, 2009. Doi: https://doi.org/10.1016/j.patrec.2008.08.010. Acesso em: 10 jan. 2026.

LIMA, Rodrigo Fonseca; CÂMARA, Tássio Fontes Moreira. Desafios com o desabastecimento na cadeia de suprimentos em um hospital público: relato de caso. *Revista de Gestão em Sistemas de Saúde*, v. 5, n. 1, p. 120–127, 2016. Doi: https://doi.org/10.5585/rgss.v5i1.218. Acesso em: 10 jan. 2026.

PINHO, Cintia Maria de Araújo; GASPAR, Marcos Antonio; SASSI, Renato José. Aplicação de técnicas de inteligência artificial para classificação de fuga ao tema em redações. *Educação em Revista*, Belo Horizonte, v. 40, e39773, 2024. Doi: http://dx.doi.org/10.1590/0102-469839773. Disponível em: https://www.scielo.br/j/edur/a/YPPVRRKGdLdFbXTQXMmZgCc/?format=html&lang=pt. Acesso em: 18 jan. 2026.

RAIMUNDO, Eliane Amâncio; DIAS, Cleidson Nogueira; GUERRA, Mariana. Logística de medicamentos e materiais em um hospital público do Distrito Federal. *Revista de Administração Hospitalar e Inovação em Saúde*, v. 12, n. 2, p. 61–69, 2015. Doi: https://doi.org/10.21450/rahis.v12i2.2384. Acesso em: 3 jan. 2026.

SOARES, Ronan Pardo. QueryAugment-RAG: aprimorando a recuperação e geração de informações em modelos de linguagem de grande escala. *Revista Contemporânea*, v. 4, n. 6, p. 1–15, 2024. Doi: 10.56083/RCV4N6-156. Acesso em: 3 jan. 2026.

YANG, Rui et al. Retrieval-Augmented Generation for Generative Artificial Intelligence in Health Care. *npj Health Systems*, v. 2, n. 2, 2025. Doi: 10.1038/s44401-024-00004-1. Disponível em: https://www.nature.com/articles/s44401-024-00004-1.pdf. Acesso em: 18 jan. 2026.