Desenvolvimento de uma Arquitetura RAG com Curadoria e Roteamento por Precedência: Aplicação em um Sistema Conversacional sobre Plantas Medicinais

Walber Amorim
Instituto Federal da Bahia
Vitória da Conquista, Brasil
walberamorim@gmail.com

Amanda Ferraz
Instituto Federal da Bahia
Vitória da Conquista, Brasil
amandainf@gmail.com

Renata Spósito
Universidade Estadual do Sudoeste da Bahia
Itapetinga, Brasil
renata.correia@uesb.edu.br

Djan Almeida
Instituto Federal da Bahia
Universidade Estadual do Sudoeste da Bahia
Vitória da Conquista, Brasil
djan.info@gmail.com

Crescencio Lima
Instituto Federal da Bahia
Vitória da Conquista, Brasil
crescencio@gmail.com

Resumo

Domínios de informação crítica em saúde exigem que sistemas conversacionais baseados em modelos de linguagem de grande porte (*Large Language Models* — LLMs) ofereçam respostas rastreáveis, seguras e auditáveis — requisitos que o uso direto de um LLM, sujeito a alucinações, não satisfaz. Este artigo apresenta e avalia um padrão arquitetural de Geração Aumentada por Recuperação (*Retrieval-Augmented Generation* — RAG) com *curadoria clínica* e *roteamento com precedência da base curada*, no qual a geração livre por LLM atua apenas como mecanismo complementar, sempre sinalizado ao usuário. A contribuição é tratada como um artefato arquitetural reutilizável: a separação explícita entre conhecimento validado, componente de recuperação semântica e componente generativo desacopla o domínio do raciocínio, favorecendo confiabilidade, rastreabilidade e substituibilidade de componentes. O padrão é instanciado em uma plataforma web para apoio ao uso de plantas medicinais no Sistema Único de Saúde (SUS), com base de conhecimento curada a partir de fontes oficiais (RENISUS, ANVISA e Farmacopeia Brasileira). A arquitetura foi submetida a um estudo empírico com 80 consultas geradas por *personas* sintéticas e comparação contra uma linha de base textual. Os resultados evidenciam que o gargalo de recuperação residia em um componente substituível (o modelo de *embedding*), e não no estilo arquitetural: ao trocar o *embedding* para um modelo multilíngue, a recuperação passou a equiparar-se à linha de base, ilustrando o valor da modularidade do padrão proposto.

Palavras-chave

Arquitetura de Software, Geração Aumentada por Recuperação, Reúso de Software, Estilo Arquitetural, Busca Semântica

1 Introdução

As práticas terapêuticas baseadas no uso de plantas medicinais e fitoterápicos são importantes componentes da Atenção Primária à Saúde (APS) em diversos países. No Brasil, a relevância dessas práticas ganha destaque no âmbito do Sistema Único de Saúde (SUS), que por sua vez busca inovação, padronização e acessibilidade plena a esses métodos e práticas que estão tão enraizados na cultura brasileira e aprimorados pela população ao longo dos anos.

Apesar de sua importância histórica e sociocultural, o uso de plantas medicinais e fitoterápicos na rede pública enfrenta alguns desafios. Entre eles, destacam-se a ausência de sistemas padronizados de informação para profissionais e usuários, a dificuldade de monitorar sua eficácia e segurança em larga escala e a persistência de práticas inadequadas de automedicação [20]. Esses fatores indicam que, embora a Política Nacional de Práticas Integrativas e Complementares (PNPIC) reconheça o potencial terapêutico das práticas, elas ainda não estão plenamente estabelecidas como uma prática estruturada no SUS.

Dados recentes reforçam essa problemática. Estudos mostram que, embora as plantas medicinais sejam amplamente utilizadas devido à sua acessibilidade e baixo custo [2, 11], sua adoção institucional ainda é dificultada por barreiras operacionais e pela carência de infraestrutura técnico-científica nos serviços de saúde [7]. Ao mesmo tempo, o avanço da fitoterapia baseada em evidências pode gerar mais equidade e sustentabilidade, valorizando a biodiversidade do país e fortalecendo práticas tradicionais [5, 6].

Nesse contexto, tecnologias emergentes, em especial aquelas baseadas em Inteligência Artificial (IA), têm se destacado como alternativas promissoras para preencher essas lacunas. Softwares dotados de funcionalidades de IA podem tornar a interação humano-tecnologia mais natural, acessível e eficiente, oferecendo suporte ao monitoramento e à gestão qualificada de informações em saúde [21]. Entretanto, o desenvolvimento de sistemas inteligentes requer o conhecimento técnico especializado para garantir a aplicabilidade correta.

Diante desse cenário, este artigo descreve o desenvolvimento de um sistema web com recursos de IA voltado ao apoio, acesso e à recuperação de informações no uso das plantas medicinais e fitoterápicos no SUS. A solução integra conhecimentos científicos e tradicionais, disponibilizando informações de forma acessível, padronizada e responsiva, por meio de interação em linguagem natural contribuindo para ampliar o acesso qualificado à informação em saúde.

Embora o domínio motivador seja a saúde pública, a contribuição central deste trabalho é *arquitetural*: descreve-se e avalia-se um padrão de software reutilizável para integração de LLMs a domínios em que a confiabilidade da informação é crítica. Concretamente, as contribuições são: (i) a caracterização do estilo arquitetural *RAG com curadoria clínica e roteamento com precedência da base curada*, no qual o conhecimento validado é desacoplado da geração livre por LLM, com discussão explícita de suas compensações (*trade-offs*) frente a alternativas como uso direto do LLM, ajuste fino (*fine-tuning*) e busca textual pura; (ii) a demonstração de que o padrão é *generalizável e reutilizável*, por separar atributos de qualidade — confiabilidade, rastreabilidade e substituibilidade de componentes — da especificidade do domínio fitoterápico; e (iii) uma avaliação empírica do artefato, baseada em *personas* sintéticas, que evidencia, entre outros achados, que o gargalo de desempenho residia em um componente substituível (o modelo de *embedding*) e não no estilo arquitetural.

2 Trabalhos Correlatos

2.1 Sistemas Conversacionais e Chatbots em Saúde

Vera and Palaoag [23] desenvolveram um protótipo de *chatbot* híbrido dotado de IA com o objetivo de responder perguntas sobre plantas medicinais e promover o uso seguro de medicação herbal. A metodologia envolveu uma avaliação quantitativa com curandeiros e usuários para medir eficácia e satisfação. Os resultados indicaram alta aceitação do sistema, destacando a utilidade de interfaces conversacionais na mediação do conhecimento tradicional.

Assim como o trabalho desenvolvido por Vera and Palaoag [23], Ahmed et al. [1] propuseram o *AyurChat*, um sistema voltado para orientação ayurvédica. Utilizando técnicas de Processamento de Linguagem Natural (PLN), o estudo focou em fornecer recomendações personalizadas baseadas em remédios tradicionais. Em contraste com o presente trabalho, que foca na realidade da saúde pública brasileira e nos desafios de inserção da fitoterapia no SUS [11, 20], o *AyurChat* é direcionado estritamente à medicina tradicional indiana.

2.2 Busca Semântica e Sistemas de Recomendação

Outra abordagem relevante foca na organização e recuperação da informação. Tran et al. [22] desenvolveram um sistema de busca baseado em uma ontologia profunda para plantas medicinais. A metodologia integrou modelos de *deep learning* para classificação de imagens com consultas ontológicas. Os resultados demonstraram que o uso de ontologias permite buscas semânticas mais precisas do que sistemas baseados apenas em palavras-chave.

Diferentemente de Tran et al. [22], que utilizam ontologias rígidas, Kumar and Kumar [13] propuseram um sistema inteligente que combina identificação automática por imagem com recomendações individualizadas. Embora eficiente na identificação visual, o sistema apresenta limitações na garantia da origem clínica dos dados, lacuna que esta pesquisa busca mitigar através de uma área de curadoria profissional.

2.3 IA e LLMs no Apoio à Engenharia de Software

Para além do domínio da saúde, o uso de IA e de LLMs como apoio ao projeto, à arquitetura e ao reúso de software tem se consolidado como linha de pesquisa ativa. Pereira et al. [17] investigaram a geração de arquiteturas de microsserviços a partir de requisitos expressos em linguagem natural, empregando LLMs para apoiar decisões arquiteturais. Na mesma direção, Freire et al. [12] aplicaram LLMs à sugestão de funções objetivo em projeto de Linhas de Produto de Software, e Sabato et al. [19] utilizaram PLN para a clusterização automática de microsserviços, evidenciando o potencial do processamento de linguagem natural em tarefas de organização arquitetural.

No tocante à avaliação dessas abordagens, Colombo et al. [8] conduziram um estudo empírico sobre o uso de LLMs para apoio à tomada de decisão na adoção de padrões de projeto, oferecendo um referencial metodológico para a avaliação de sistemas assistidos por IA — enfoque que orienta o design de avaliação adotado neste trabalho. Cabe ainda destacar que arquiteturas de software voltadas ao domínio da saúde também têm sido objeto de investigação, como em Pohlmann et al. [18], que propuseram uma solução de comunicação adaptável para tráfego de dados clínicos em ambientes *Edge-Fog-Cloud*.

Diante desse panorama, o presente trabalho situa-se na interseção entre essas frentes: aplica o padrão arquitetural de RAG, com curadoria clínica e busca semântica vetorial, a um problema concreto de saúde pública, contribuindo com uma arquitetura reutilizável para domínios em que a confiabilidade da informação é crítica.

2.4 Lacuna Identificada

Ao analisar os trabalhos correlatos, observa-se que a maioria foca em identificação por imagem ou em medicinas tradicionais estrangeiras. Persiste a carência de sistemas que integrem a flexibilidade dos LLMs com bases de dados baseadas em documentos oficiais do SUS, dentre eles a Relação Nacional de Plantas Medicinais de Interesse ao SUS (RENISUS)), farmacopeia e monografias. Esta pesquisa busca preencher tal lacuna ao utilizar a arquitetura `pgvector` para garantir que a informação recuperada priorize registros validados e seguros. A Tabela 1 apresenta a comparação entre os trabalhos correlatos e o sistema proposto.

3 Metodologia

A estratégia metodológica consistiu na implementação de um software web baseado no modelo de interação conversacional. O processo contemplou a análise de requisitos, definição da arquitetura, escolha das tecnologias e implementação das funções centrais do sistema. A coleta dos dados sobre as plantas medicinais foi conduzida por uma equipe de pesquisadores com experiência na área.

O desenvolvimento da plataforma foi conduzido em quatro etapas principais: (1) levantamento e especificação de requisitos, (2) definição da arquitetura de software, (3) implementação da aplicação web e serviços *backend*, e (4) integração dos componentes de IA conversacional e busca semântica, seguida de validação por testes.

Tabela 1: Comparação entre os trabalhos correlatos e o sistema proposto.

Trabalho	Abordagem	Int. Conv.	LLM/IA	Curadoria Clínica	RAG	Contexto SUS
Vera and Palaoag [23]	Chatbot híbrido para plantas medicinais	Sim	Parcial	Não	Não	Não
Ahmed et al. [1]	AyurChat: PLN para orientação ayurvédica	Sim	Parcial	Não	Não	Não
Tran et al. [22]	Busca ontológica com *deep learning*	Não	Sim	Não	Não	Não
Kumar and Kumar [13]	Identificação por imagem com recomendação individualizada	Parcial	Sim	Não	Não	Não
Este trabalho	RAG + LLM com curadoria clínica voltada ao SUS	Sim	Sim	Sim	Sim	Sim

3.1 Materiais e ferramentas utilizados

Para o desenvolvimento do sistema, foram utilizadas as IDEs e editores de código Visual Studio Code e IntelliJ IDEA para codificação do projeto. A aplicação foi implementada utilizando a linguagem TypeScript, estruturada no *frontend* com o framework Angular 20. No *backend*, utilizou-se a linguagem Java 17, com a organização das rotas e lógicas de negócio apoiadas no framework Spring Boot 3.5.6. O armazenamento de dados ocorreu no banco relacional PostgreSQL, com suporte da extensão `pgvector` para indexação e consultas a dados vetoriais. Para a integração das funcionalidades de IA, empregou-se a API da plataforma Google Gemini, utilizando o modelo gratuito gemini-2.5-flash.

3.2 Composição da Base de Conhecimento

A base de conhecimento do sistema foi estruturada a partir de fontes oficiais do SUS, incluindo a RENISUS, monografias da Agência Nacional de Vigilância Sanitária (ANVISA) e a Farmacopeia Brasileira. A curadoria dos registros foi conduzida por um especialista. Para a validação técnica descrita neste trabalho, a base continha plantas medicinais aprovadas. Cada registro inclui os campos de indicações terapêuticas, contraindicações, modo de uso, compostos bioativos e evidências de atividade antimicrobiana.

4 Desenvolvimento

A proposta foi concebida como um sistema web interativo dividido em duas áreas: uma interface pública baseada em *chatbot* e um ambiente privado restrito para manutenção do banco de dados. A interface de *chatbot* utiliza diálogos em linguagem natural e consulta uma base de plantas medicinais previamente validada, além de empregar busca semântica para recuperar informações clínicas de forma contextual. O ambiente privado disponibiliza formulários para o cadastro e gerenciamento das plantas medicinais na base de dados, permitindo a ampliação do repositório de forma estruturada. Nesse sistema foi estabelecida uma hierarquia simples para os usuários, onde o usuário gerencial (gerente) teria o papel de inserir as plantas para análise somente, e o administrador seria capaz de realizar todas as atividades privadas do sistema, incluindo a aprovação das plantas para que as mesmas sejam adicionadas à base de dados curada do sistema.

4.1 Requisitos do Sistema

O sistema foi desenvolvido considerando um conjunto de requisitos funcionais (RF) e não funcionais (RNF) alinhados às necessidades de interação em linguagem natural, curadoria profissional da base de plantas medicinais e armazenamento otimizado para consultas assistidas por IA. Esses requisitos orientam tanto a experiência do usuário final, mediada pelo *chatbot*, quanto os fluxos de validação e gestão de dados no ambiente restrito.

A Tabela 2 apresenta os principais requisitos funcionais e não funcionais usados como base para o projeto.

4.2 Arquitetura do Sistema

A solução adota uma arquitetura de aplicação web dividida em *frontend* e *backend*, com fluxo de dados mediado por APIs REST documentadas. O *frontend* foi implementado como uma *Single Page Application* (SPA) utilizando o framework Angular, responsável pelas telas de login, formulários de submissão de plantas medicinais, listagens e a interface livre do *chatbot* (sem necessidade de autenticação). O *backend* foi desenvolvido em Java com o framework Spring Boot, integrando mecanismos de recuperação contextual e processamento de linguagem natural por meio da biblioteca `LangChain4j 1.7.1`. O banco de dados relacional utilizado foi o PostgreSQL, versionado e gerenciado via `Liquibase`, incluindo suporte a armazenamento vetorial com a extensão `pgvector` para indexação semântica das plantas aprovadas. Para respostas externas auxiliares, integrou-se a API do modelo gemini-2.5-flash. A documentação interativa da API é disponibilizada através do `Swagger UI`.

A Figura 1 apresenta a arquitetura geral da plataforma desenvolvida, evidenciando a separação entre *frontend*, *backend*, banco de dados e serviço externo de IA (API Gemini).

A Figura 2 detalha o fluxo de processamento de uma consulta. A decisão de roteamento, privilegiar a base curada e recorrer ao Gemini apenas quando a similaridade recuperada não atinge o limiar mínimo, constitui a principal decisão arquitetural da solução, garantindo precedência às informações clínicas validadas.

Tabela 2: Principais requisitos funcionais e não funcionais do sistema

ID	Descrição do Requisito	Classificação
RF1	Processar perguntas realizadas em linguagem natural e gerar respostas estruturadas com base em registros previamente validados por profissionais de saúde.	Funcional
RF2	Fornecer informações sobre indicações terapêuticas, contraindicações, modos de uso, compostos bioativos e evidências de atividade antimicrobiana das plantas cadastradas.	Funcional
RF3	Permitir submissão de novas plantas por meio de formulário para avaliação e aprovação do administrador antes da inclusão definitiva na base de dados do sistema.	Funcional
RF4	Exibir listagens de plantas aprovadas e de submissões em análise, apresentando o status atualizado do processo de curadoria.	Funcional
RNF1	Armazenar plantas aprovadas em banco de dados vetorial, possibilitando consultas semânticas e recuperação contextual assistida por IA.	Não Funcional
RNF2	Acionar respostas geradas por agente auxiliar de IA apenas quando não houver correspondência suficiente na base validada, informando explicitamente ao usuário a origem desse retorno.	Não Funcional

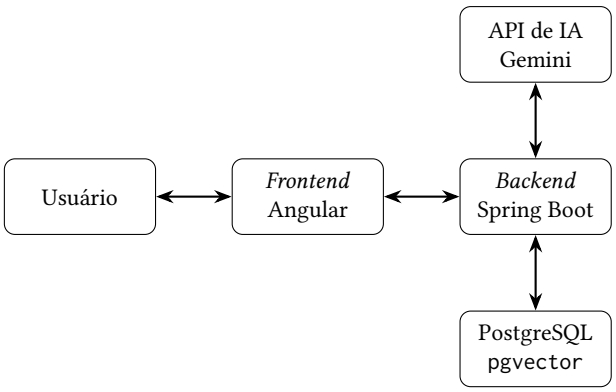

Figura 1: Arquitetura geral da plataforma desenvolvida

4.3 RAG como Estilo Arquitetural: Compensações e Reuso

Mais do que uma escolha de implementação, a solução instancia um *estilo arquitetural* para sistemas conversacionais sobre conhecimento crítico. O RAG [9, 14] decompõe o sistema em quatro elementos de primeira classe: (i) uma base de conhecimento curada, fonte de verdade do domínio; (ii) um componente de recuperação que projeta consultas e documentos em um mesmo espaço vetorial e recupera registros por similaridade; (iii) um roteador, que decide se a consulta é atendida pela base curada ou encaminhada à geração; e (iv) um componente generativo (LLM), acionado apenas como mecanismo complementar. Os conectores entre esses elementos são bem definidos — APIs REST entre cliente e serviço, e operadores de similaridade vetorial entre recuperação e base —, o que confere ao estilo um caráter modular e instanciável em diferentes domínios.

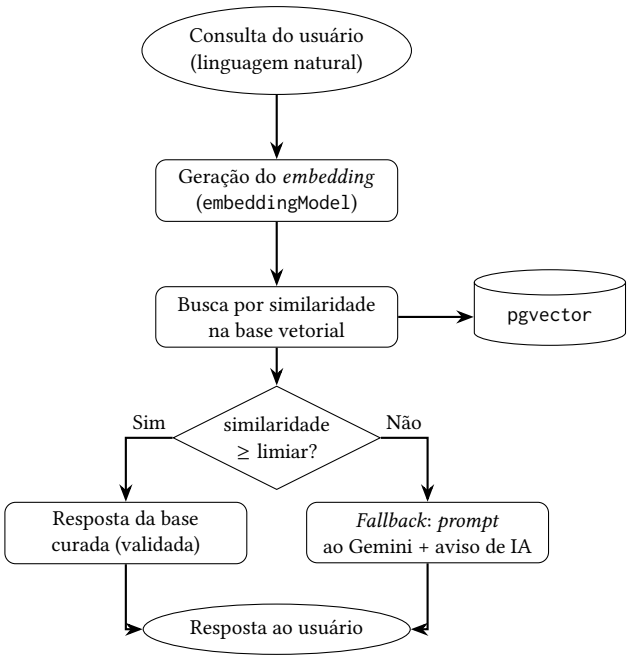

Figura 2: Fluxo de processamento da consulta.

A particularização adotada neste trabalho é a *precedência da base curada* (*curated-first*): a geração livre por LLM só é acionada quando a recuperação não satisfaz o critério de relevância, e toda resposta gerada é explicitamente sinalizada como tal. Essa decisão arquitetural subordina a flexibilidade do LLM à rastreabilidade da informação clínica, atacando diretamente o risco de alucinação em um domínio sensível.

4.3.1 Compensações frente a alternativas. A escolha do estilo decorre de compensações (*trade-offs*) explícitas entre atributos de qualidade.

O uso direto do LLM oferece a interação mais fluida, mas não há proveniência: a resposta não pode ser rastreada a uma fonte validada e o risco de alucinação é elevado — inaceitável em saúde. O ajuste fino internaliza o domínio nos pesos do modelo, porém torna a atualização custosa (cada revisão da base exige novo treinamento), opaca (não se distingue conhecimento validado de inferência) e sem proveniência explícita. A busca textual pura maximiza rastreabilidade e elimina alucinação, mas não gera linguagem natural e é frágil a sinônimos, nomes populares e erros de digitação, frequentes entre os usuários-alvo. O estilo RAG *curated-first* equilibra esses extremos: preserva a proveniência da busca, mantém baixo o custo de atualização (basta editar a base curada) e recupera a fluência do LLM, restringindo a geração livre a um papel complementar e sinalizado.

4.3.2 Generalização e reuso. A reutilização do artefato decorre de o estilo separar *atributos de qualidade* da *especificidade do domínio*. Os requisitos endereçados — confiabilidade da informação, rastreabilidade da resposta à fonte e substituibilidade de componentes — não dependem do conteúdo fitoterápico; aplicam-se a qualquer domínio de saúde em que respostas devam ser auditáveis (bulas e interações medicamentosas, protocolos clínicos, orientações de vacinação). Reusar o padrão em um novo domínio exige, essencialmente, substituir a base de conhecimento curada e, quando necessário, o modelo de *embedding* adequado ao idioma e ao vocabulário, sem alterar a estrutura de roteamento nem os conectores. A modularidade não é apenas conceitual: como evidenciado na Seção 6, a substituição isolada do componente de *embedding* alterou substancialmente a qualidade da recuperação sem qualquer mudança no restante da arquitetura, confirmando empiricamente a substituibilidade que o estilo promete.

4.4 Implementação e Fluxo de Funcionamento

A implementação do sistema segue um fluxo de responsabilidades baseado em perfis de acesso e recuperação semântica de informações. O acesso inicial ocorre por meio do módulo de autenticação do *frontend*, que disponibiliza a tela de login e controle de sessão.

Após autenticado, o usuário acessa o painel privado, submete uma nova planta medicinal por meio de formulários estruturados (contendo informações como indicações, contraindicações, modo de uso e compostos bioativos) e essa submissão é registrada no banco relacional com *status* pendente de análise.

A Figura 3 apresenta a interface do formulário responsável pela submissão das plantas para análise, contendo função de autocompletar no campos de "Principais compostos" e "Atividade antimicrobiana" caso a informação inserida já esteja cadastrada na base de dados do sistema.

O administrador, autenticado em perfil mais elevado, acessa o painel de gestão de submissões, onde são disponibilizadas listagens das submissões pendentes. A Figura 4 apresenta a listagem de submissões pendentes para aprovação.

Ao aprovar um registro, o sistema persiste os dados permanentemente em tabelas convencionais e, em seguida, gera a representação vetorial da planta utilizando a biblioteca LangChain4j. Nesse processo, o modelo de embeddings (`embeddingModel`) é acionado,

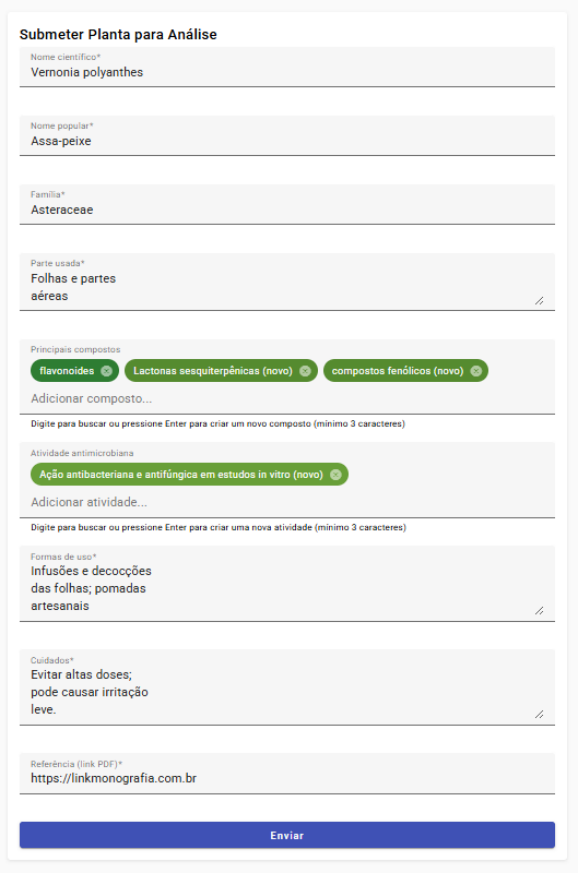

Figura 3: Tela de submissão da planta para análise

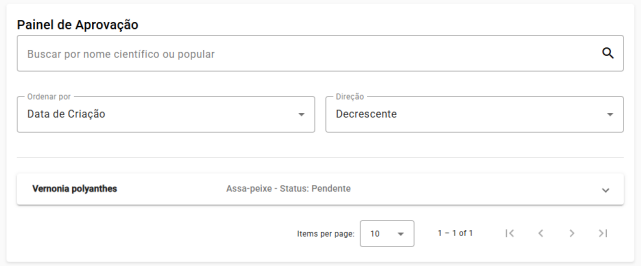

Figura 4: Tela de aprovação de submissões

converte os atributos textuais da planta em vetor denso e o insere na tabela vetorial (`pgvector`), permitindo consultas por similaridade semântica. Na Figura 5 é apresentado o trecho de código responsável pela manipulação das informações da planta aprovada e geração do dado em formato vetorial.

Na interface pública do *chatbot*, o usuário final pode enviar perguntas em linguagem natural, que são processadas e convertidas em vetor de consulta pelo mesmo `embeddingModel`. A busca é então realizada na base vetorial por similaridade (ex: `cosine similarity` ou `inner product`), e, caso a recuperação retorne um valor acima do limiar mínimo de relevância, o sistema formata e exibe os dados

```
String textToEmbed = textBuilder.toString();
Embedding embedding = embeddingModel
                      .embed(textToEmbed)
                      .content();
float[] vector = embedding.vector();

PlantEmbedding plantEmbedding = PlantEmbedding
                                .builder()
                                .plant(plant)
                                .embedding(vector)
                                .createdAt(LocalDateTime.now())
                                .build();
```

Figura 5: Processo de obtenção do texto e geração do embedding vetorial

clínicos vinculados à planta mais similar. Na Figura 6 é apresentado como a consulta é feita para recuperar as informações na base vetorial. A Figura 7 apresenta a interface do chat obtendo a resposta com os dados originados da base de dados cadastrada e aprovada pelos usuários.

```
@Query(value = "SELECT pe.id, pe.plant_id,
    string_to_array(substring(pe.embedding::text
    from 2 for length(pe.embedding::text)-2), ',')::float[]
    as embedding,
    pe.created_at
FROM plant_embeddings pe
ORDER BY pe.embedding <-> CAST(:queryVector AS vector)
LIMIT :limit", nativeQuery = true)
List<PlantEmbedding> findNearestNeighbors(@Param("queryVector")
    float[] queryVector, @Param("limit") int limit);
```

Figura 6: Consulta SQL com uso de similaridade vetorial para recuperação dos registros mais próximos

Quando nenhuma correspondência semântica suficiente é encontrada, o agente auxiliar aciona a API externa do modelo gemini-2.5-flash, enviando um *prompt* especializado contendo instruções de resposta estruturada e segura, recebendo o retorno gerado pelo modelo gemini-2.5-flash. Na Figura 8 são listadas as instruções de *prompt* passadas para o modelo escolhido com o objetivo de obter uma resposta que tenha a mesma estrutura das respostas criadas a partir da base de dados do sistema. É destacado também que aquela resposta foi gerada por uma IA e não pela base de dados curada.

Nesse caso, a resposta é exibida ao usuário acompanhada de um aviso automático indicando que o conteúdo foi produzido por um agente auxiliar de IA e não por registros curados pela equipe clínica.

Esse fluxo de implementação — login, submissão, curadoria profissional, vetorização e recuperação semântica — garante que as respostas priorizem dados embasados em evidências e validados pelo SUS e Anvisa, utilizando a geração externa apenas como mecanismo complementar quando necessário.

5 Resultados

Esta seção apresenta os resultados obtidos a partir do desenvolvimento e da avaliação técnica da plataforma proposta. Os resultados concentram-se na validação do funcionamento do sistema, na integração entre seus componentes e na capacidade de resposta às

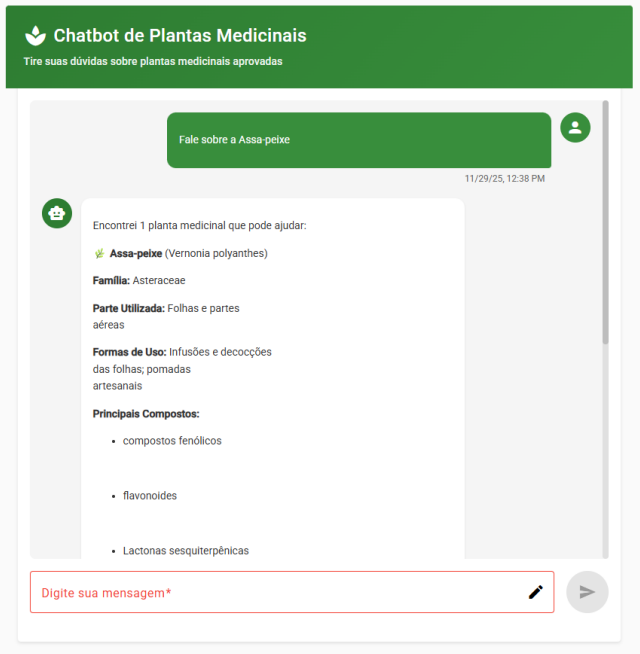

Figura 7: Resposta vinda da base de dados

```
@SystemMessage("""
Você é um especialista em plantas medicinais brasileiras com conhecimento profundo em fitoterapia.
Ao responder sobre plantas medicinais, forneça informações estruturadas no seguinte formato:

**Nome Popular e Científico**

**Família Botânica:** [se souber]

**Benefícios para a Saúde:**
• Liste os principais benefícios terapêuticos
• Base as informações em conhecimento científico

**Formas de Uso:**
• Chá/infusão
• Decocção
• Outras formas tradicionais

**Principais Compostos:** [se relevante]
• Liste compostos bioativos importantes

**Precauções e Contraindicações:**
• Grupos que devem evitar (grávidas, crianças, etc.)
• Possíveis efeitos colaterais
• Interações medicamentosas conhecidas

Use linguagem clara e acessível, mas mantenha rigor científico.
Se não tiver certeza sobre alguma informação, indique isso.
Caso a pergunta envolva assuntos fora do escopo de plantas medicinais,
responda educadamente que não pode ajudar com esse tema.
""")
```

Figura 8: Prompt utilizado para fornecer contexto ao modelo Gemini na geração das respostas

interações realizadas em linguagem natural. Ressalta-se que não foi realizada avaliação clínica ou terapêutica, uma vez que o objetivo esteve restrito à implementação da infraestrutura tecnológica.

5.1 Funcionamento geral da plataforma

A plataforma desenvolvida resultou em uma aplicação web funcional, acessível por navegadores, organizada em arquitetura cliente-servidor. O *frontend* permitiu a interação do usuário por meio de uma interface simples e responsiva, enquanto o *backend* foi responsável pelo processamento das requisições, comunicação com o banco de dados e integração com os serviços de IA.

5.2 Integração da Plataforma com IA

A integração com o serviço de IA permitiu que a plataforma interpretasse consultas realizadas em linguagem natural. Os testes demonstraram que o sistema foi capaz de identificar a intenção do usuário e gerar respostas coerentes com base nas informações previamente indexadas no banco de dados vetorial.

Observou-se que o uso de indexação vetorial contribuiu para a recuperação eficiente de informações relevantes, reduzindo o volume de texto enviado ao modelo de linguagem e favorecendo respostas mais contextualizadas. Mesmo utilizando um modelo gratuito de IA, os resultados obtidos foram satisfatórios para o propósito da aplicação.

5.3 Avaliação de Qualidade com Google Lighthouse

Para avaliar a qualidade técnica das páginas da aplicação, foram coletadas métricas por meio da ferramenta Google Lighthouse (versão 13.0.2), que analisa aspectos de Desempenho, Acessibilidade, Boas Práticas e Otimização para Mecanismos de Busca (SEO). A ferramenta também fornece métricas de *Web Vitals*: *First Contentful Paint* (FCP), *Largest Contentful Paint* (LCP), *Total Blocking Time* (TBT) e *Cumulative Layout Shift* (CLS). As cinco páginas principais da plataforma foram avaliadas e os resultados são apresentados na Tabela 3.

Os resultados evidenciam elevado desempenho técnico da plataforma, com média de 95,6/100 na categoria Desempenho. A página de submissão de plantas (*/submit*) e a do Chatbot atingiram pontuação máxima de 99/100, sendo que */submit* obteve ainda pontuação perfeita em Boas Práticas (100/100). O menor índice de Desempenho foi observado na página Minhas Submissões (91/100), associado ao maior Tempo Total de Bloqueio (TBT = 230 ms). Em Acessibilidade, a página do Chatbot obteve 86/100, valor ligeiramente abaixo das demais (94/100), reflexo dos desafios inerentes à interface conversacional dinâmica. A pontuação de SEO foi uniforme (73/100) em todas as páginas, motivada principalmente pela ausência de metadados descritivos e pela configuração incompleta do arquivo `robots.txt`, aspectos previstos para versões futuras da plataforma.

6 Avaliação Empírica

Para avaliar o artefato arquitetural além da verificação funcional da Seção 5, conduziu-se um estudo empírico sobre a qualidade da recuperação, a efetividade do roteamento e a segurança das respostas [25] e, por envolver um sistema baseado em LLM, alinhado às diretrizes específicas para estudos empíricos com LLMs [3]. A organização do estudo seguiu o paradigma *Goal–Question–Metric* (GQM) [4] e optou-se por um estudo de simulação baseado em *personas* sintéticas, que permite uma avaliação sistemática e reprodutível sem expor o sistema a participantes humanos nesta etapa pré-piloto: as consultas são geradas a partir de perfis de usuário sintéticos e confrontadas automaticamente com um conjunto rotulado e congelado, do qual derivam todas as métricas das RQ1–RQ5.

6.1 Objetivo, Perguntas e Métricas (GQM)

Objetivo (Goal). *Analisar* o padrão arquitetural RAG com curadoria clínica e roteamento com precedência da base curada, *com o propósito de* avaliá-lo (caracterizá-lo e validá-lo), *com respeito à* eficácia da recuperação, à efetividade do roteamento, à robustez, à necessidade da abordagem e ao desempenho, *do ponto de vista do* pesquisador em Engenharia de Software (e, indiretamente, do profissional de saúde usuário), *no contexto de* um sistema-piloto de apoio ao uso de plantas medicinais no SUS, avaliado por simulação com *personas* sintéticas.

Perguntas (Questions). O objetivo desdobra-se em cinco perguntas de pesquisa (RQ1–RQ5):

- **RQ1 (Recuperação):** Com que precisão o sistema recupera o registro de planta correto a partir de uma consulta em linguagem natural?
- **RQ2 (Roteamento):** O critério de roteamento separa adequadamente as consultas respondíveis pela base curada daquelas que exigem o mecanismo generativo complementar?
- **RQ3 (Robustez):** Como o sistema se comporta diante de paráfrases, nomes populares e erros de digitação?
- **RQ4 (Necessidade da abordagem):** A recuperação semântica (RAG) supera uma busca textual convencional sobre o mesmo corpus?
- **RQ5 (Desempenho):** Qual a latência de resposta em cada caminho (curado e *fallback*)?

Métricas (Metrics). A Tabela 4 mapeia cada pergunta às suas métricas, detalhadas a seguir.

RQ1 — Hit@k e MRR. Hit@k (ou *Hits@k*) é a fração de consultas em que o registro correto aparece entre as k primeiras posições do *ranking*: $\text{Hit@}k = \frac{1}{N}\sum_{i=1}^{N} \mathbb{1}[\text{rank}_i \le k]$, com N consultas e rank_i a posição do alvo na consulta i [15]. Reportam-se Hit@1 (acerto na primeira posição) e Hit@3 (alvo no *top-3*). O *Mean Reciprocal Rank* (MRR) pondera a posição pelo seu inverso, $\text{MRR} = \frac{1}{N}\sum_{i=1}^{N} \frac{1}{\text{rank}_i}$, premiando respostas corretas mais no topo [24]; varia de 0 a 1. Enquanto Hit@k é binário dentro do corte, o MRR diferencia a qualidade do ordenamento.

RQ2 — Precisão, revocação, F1 e AUC. A classificação curado *vs. fallback* é avaliada por precisão (fração de roteamentos ao caminho curado que estavam corretos), revocação (fração das consultas verdadeiramente curadas que foram roteadas como tal) e F1 (média harmônica entre ambas) [15]. Como o roteador em produção é léxico (sem limiar), conduz-se também uma análise *offline* por limiar de similaridade, sumarizada pela AUC-ROC — área sob a curva ROC, que mede a capacidade de discriminação independentemente do limiar — e pelo melhor F1 ao longo dos limiares [10].

RQ3 — Consistência *top-1*. A robustez a variações lexicais é medida pela taxa de consistência *top-1*: proporção de paráfrases (categoria D) que recuperam o mesmo registro *top-1* da consulta canônica correspondente.

RQ4 — Teste de McNemar. A comparação entre RAG e *baseline* textual sobre os mesmos pares de consultas usa o teste pareado de McNemar [16], apropriado para dados emparelhados de acerto/erro, com nível de significância $\alpha = 0{,}05$.

RQ5 — Latência. Reportam-se a latência mediana e o percentil 95 (p95) do tempo de resposta em cada caminho, métricas

Tabela 3: Avaliação de qualidade das páginas com Google Lighthouse (versão 13.0.2).

Página	Desempenho	Acessibilidade	Boas Práticas	SEO	FCP	LCP	TBT	CLS
Painel de Aprovação	93	94	96	73	0,6 s	0,8 s	200 ms	0,002
Plantas Aprovadas	96	94	96	73	0,7 s	0,7 s	160 ms	0,002
Chatbot	99	86	96	73	0,8 s	0,9 s	40 ms	0,000
Minhas Submissões	91	94	96	73	0,7 s	0,8 s	230 ms	0,002
Submeter Planta	99	94	100	73	0,6 s	0,7 s	90 ms	0,000
Média	**95,6**	**92,4**	**96,8**	**73,0**	**0,68 s**	**0,78 s**	**144 ms**	**0,001**

Tabela 4: Mapa GQM: objetivo → perguntas → métricas.

Pergunta	Métricas associadas
RQ1	Hit@1, Hit@3, MRR [15, 24]
RQ2	Precisão, revocação, F1 [15]; AUC-ROC e melhor F1 por limiar [10]
RQ3	Taxa de consistência *top-1* sob paráfrase
RQ4	Teste pareado de McNemar [16] sobre acertos *top-1* (RAG *vs. baseline*)
RQ5	Latência mediana e percentil 95 (p95) por caminho

usuais de desempenho por serem robustas a *outliers* e representativas da experiência típica e de cauda.

6.2 Design do Estudo

A Figura 9 apresenta o design do estudo. Para evitar a circularidade em que o sistema avaliaria a si mesmo, os três papéis do estudo foram atribuídos a *provedores distintos* (Tabela 5): nenhum modelo gera, responde e julga ao mesmo tempo, eliminando o viés de autopreferência. As versões exatas dos modelos são registradas para fins de reprodutibilidade.

O desenho operacionaliza as diretrizes de Baltes et al. [3] para estudos empíricos com LLMs: declara-se o uso de LLM e seu papel no sistema; reportam-se as versões e a configuração dos modelos (Tabelas 5 e 6), incluindo *embedding*, índice vetorial e critério de roteamento; emprega-se uma linha de base não baseada em LLM (busca textual) com métricas de recuperação e de classificação adequadas (RQ1, RQ2, RQ4); registra-se o conjunto de consultas de forma congelada e versionada (*hash* SHA-256), permitindo reexecução; e as limitações e respectivas mitigações são explicitadas na Seção 6.6. Como ponto em aberto frente a essas diretrizes, não se incluiu nesta etapa-piloto uma validação sistemática das saídas por especialista humano nem um modelo de LLM aberto como comparação, ambos registrados como ameaças à validade e trabalho futuro.

6.3 Conjunto de Avaliação e Personas

O conjunto foi gerado a partir de cinco *personas* sintéticas — médico(a) da APS, farmacêutico(a), enfermeiro(a)/ACS, usuário leigo e uma

Tabela 5: Separação de papéis por provedores distintos (anti-circularidade).

Papel	Provedor / modelo
Gerador de consultas	Cohere — `command-a-03-2025`
Sistema sob avaliação	Google — `gemini-2.5-flash` (pipeline RAG)

persona adversária —, cada qual com vocabulário e intenções próprios. As consultas foram rotuladas em quatro categorias, cujo gabarito constitui a referência para o roteamento e a robustez: (A) na base curada (espera-se caminho curado); (B) fora da base (espera-se *fallback*); (C) *adversarial*/insegura (espera-se recusa ou alerta); e (D) paráfrases de (A) com sinônimos, nomes populares ou erros de digitação. O conjunto totaliza 80 consultas (5 *personas* × 4 categorias × 4). Para mitigar viés de confirmação, ele foi congelado antes da execução e registrado com *hash* de integridade SHA-256 , funcionando como pré-registro.

A linha de base (RQ4) é uma busca textual *full-text* do PostgreSQL (`tsvector`/`ts_rank`) sobre o mesmo corpus curado, sem LLM, isolando o ganho atribuível à recuperação semântica. Os parâmetros do sistema avaliado, exigidos para reprodutibilidade, constam na Tabela 6.

Tabela 6: Parâmetros do sistema avaliado.

Parâmetro	Valor
Plantas na base curada	12 (conjunto-piloto, validado por especialista)
Modelo de *embedding*	`paraphrase-multilingual MiniLM-L12-v2` (ONNX, 384-d)
Métrica de similaridade	cosseno ($1 - \langle = \rangle$)
Índice `pgvector`	IVFFlat (`vector_cosine_ops`)
Roteamento	léxico (casamento de nome), não por limiar
LangChain4j	`1.7.1-beta14`
Modelo generativo	`gemini-2.5-flash`

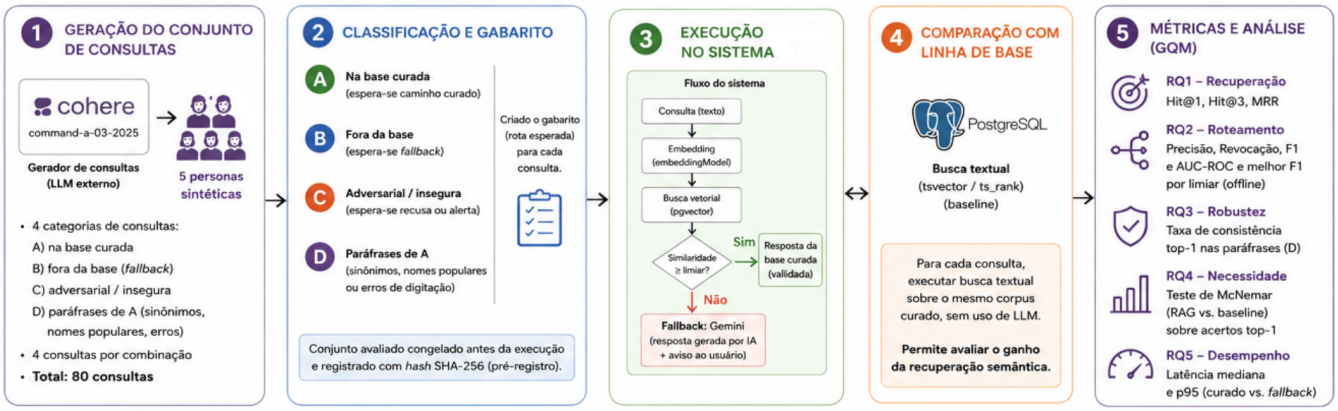

Figura 9: Design do Estudo

6.4 Resultados

A seguir apresentam-se os resultados da avaliação empírica.

RQ1 — Recuperação. Nas 40 consultas com alvo conhecido (categorias A e D), o pipeline RAG atingiu Hit@1 de 0,525, Hit@3 de 0,650 e MRR de 0,610, valores próximos aos da linha de base textual (Hit@1 0,575; Hit@3 0,675; MRR 0,636), como mostra a Tabela 7.

Tabela 7: RQ1 — recuperação (categorias A+D, $N = 40$).

Sistema	Hit@1	Hit@3	MRR
RAG (proposto)	0,525	0,650	0,610
Baseline textual	0,575	0,675	0,636

RQ2 — Roteamento. O roteador léxico em produção obteve precisão de 0,543, revocação de 0,950 e F1 de 0,691 ao distinguir consultas A (curado) de B (*fallback*). Como o roteamento atual não emprega limiar de similaridade, a análise por limiar foi conduzida *offline* sobre a similaridade registrada por consulta: a curva ROC apresentou AUC $\approx$ 0,720, com melhor F1 de 0,741 no limiar $t = 0{,}460$, indicando que um roteador por limiar é viável e derivável dos dados coletados.

RQ3 — Robustez. Das 20 paráfrases (categoria D), 60,0% (12/20) recuperaram o mesmo registro *top-1* da consulta canônica correspondente, indicando sensibilidade moderada a variações lexicais.

RQ4 — Necessidade da abordagem. No teste pareado de McNemar sobre Hit@1 (40 pares), o RAG acertou isoladamente em 4 casos e a linha de base em 6, sem diferença estatisticamente significativa ($p = 0{,}754$). Ou seja, com o *embedding* multilíngue, a recuperação semântica equipara-se à busca textual, não a supera neste conjunto-piloto.

RQ5 — Desempenho. A latência mediana foi de 61 ms (p95 = 161 ms) no caminho curado e de 11 905 ms (p95 = 23 216 ms) no *fallback* — cerca de **195**× mais lento na mediana, refletindo o custo direto da chamada externa ao provedor generativo.

6.5 O Componente de *Embedding* como Gargalo: Evidência de Substituibilidade

O achado mais relevante para a perspectiva arquitetural emergiu de uma análise de variável controlada em que apenas o modelo de *embedding* foi alterado, mantendo-se idênticos texto, base, consultas congeladas, linha de base e roteador. A configuração inicial usava um *embedding em inglês* (`MiniLM-L6-v2`) sobre um corpus em português; a troca por um modelo *multilíngue* de mesma dimensão (`vector(384)`, *drop-in*) alterou drasticamente os resultados (Tabela 8): Hit@1 subiu de 0,050 para 0,525, o MRR de 0,258 para 0,610, a discriminação do roteamento (AUC) de 0,435 para 0,720, e o RAG, que antes *perdia* para a linha de base ($p < 0{,}001$), passou a empatar com ela ($p = 0{,}75$).

O gargalo, portanto, não estava no *estilo arquitetural*, mas em um *componente substituível*. Esse resultado valida empiricamente o atributo de qualidade central do padrão proposto na Seção 4.3: a substituibilidade de componentes permitiu corrigir a recuperação sem qualquer alteração na base, no roteador ou nos conectores. Um *probe* offline com um *embedding* mais forte (Cohere v3, Hit@1 0,750) sugere ainda margem para que o RAG *supere* a linha de base, indicando que a evolução do sistema é uma questão de troca de componente, não de redesenho.

6.6 Ameaças à Validade

Durante o desenvolvimento e a avaliação técnica da plataforma, foram identificadas ameaças que podem comprometer a estabilidade e a fidedignidade dos resultados, exigindo estratégias específicas de mitigação.

Limitações da API de IA. A principal ameaça técnica reside na dependência de serviços externos, especificamente a utilização do modelo Google Gemini 2.5 Flash em sua versão gratuita. Esta modalidade impõe restrições quanto ao volume de tokens processados e à necessidade de manter uma conexão estável com provedores externos. *Mitigação:* Para contornar essa limitação, o sistema utiliza a extensão `pgvector` no PostgreSQL para realizar buscas semânticas locais. Essa abordagem reduz drasticamente o volume de texto enviado ao modelo de linguagem, uma vez que a API externa é acionada apenas como mecanismo complementar quando não há

Tabela 8: Efeito da substituição isolada do modelo de *embedding* (mesmo conjunto congelado de 80 consultas).

Métrica	MiniLM-L6 (inglês)	MiniLM-L12 (multilíngue)	Cohere v3 (*probe* offline)	*Baseline* textual
RQ1 Hit@1	0,050	**0,525**	0,750	0,575
RQ1 MRR	0,258	**0,610**	0,824	0,636
RQ2 AUC (roteamento)	0,435	**0,720**	—	—
RQ4 RAG vs. *baseline*	$p < 0{,}001$ (perde)	$p = 0{,}75$ **(empata)**	—	—

correspondência suficiente na base local curada, favorecendo respostas mais contextualizadas e economizando recursos de tokens.

Confiabilidade e Acurácia das Informações. Uma ameaça crítica diz respeito à geração de respostas que possam conter imprecisões técnicas caso a base de dados não esteja devidamente alimentada ou o modelo de IA utilizado sofra "alucinações". A carência de sistemas padronizados no SUS torna a segurança da informação um ponto sensível. *Mitigação:* Para garantir que a informação recuperada priorize registros validados e seguros, a arquitetura `pgvector` foi configurada para dar precedência absoluta à base de dados clínica. Além disso, foi implementado um *prompt* especializado (*System Message*) que instrui o modelo Gemini a manter rigor científico e indicar incertezas. Como medida de transparência, o sistema exibe um aviso automático sempre que uma resposta é produzida pelo agente auxiliar de IA, distinguindo-a claramente dos registros curados por especialistas.

Validade de construto. As *personas* sintéticas não substituem usuários reais; o estudo posiciona-se como avaliação técnica pré-piloto, não como estudo de usabilidade.

Validade estatística e de conclusão. Trata-se de um estudo-piloto (12 plantas, 80 consultas; *fallback* $N = 11$), de modo que as taxas do *fallback* têm baixo poder. Ampliar a base e o número de consultas é necessário para aumentar o poder, sobretudo das categorias A e D. O roteador real é léxico; a análise ROC por limiar é *offline* e indica o comportamento de um roteador por limiar, não o do roteador atualmente em produção.

Validade externa. A base-piloto e as *personas* são específicas de plantas medicinais (PT-BR); a generalização a outros domínios e idiomas não foi testada.

7 Conclusão

Este artigo caracterizou e avaliou um padrão arquitetural reutilizável — RAG com curadoria clínica e roteamento com precedência da base curada — para a integração de LLMs a domínios em que a confiabilidade da informação é crítica. A contribuição central não é a aplicação fitoterápica em si, mas o estilo arquitetural que a sustenta: ao desacoplar o conhecimento validado, a recuperação semântica e a geração livre, o padrão promove confiabilidade, rastreabilidade da resposta à fonte e substituibilidade de componentes, atributos discutidos frente a alternativas como uso direto do LLM, ajuste fino e busca textual pura (Seção 4.3). O padrão foi instanciado em uma plataforma web funcional, com arquitetura SPA e *backend* REST, validada quanto à qualidade de software.

A avaliação empírica sustenta essas afirmações com evidências, em vez de generalizações. O achado mais expressivo do ponto de vista de Engenharia de Software é que o gargalo de recuperação residia em um *componente substituível* (o modelo de *embedding*, inadequado ao idioma) e não no estilo arquitetural: sua troca isolada elevou o Hit@1 de 0,05 para 0,525 sem qualquer alteração na base, no roteador ou nos conectores — uma demonstração concreta da modularidade pretendida. Com o *embedding* adequado, a recuperação semântica passou a **equiparar-se** à linha de base textual ($p = 0{,}75$), e um *probe* offline indicou margem para superá-la.

Reconhecem-se limitações que delimitam o alcance dos resultados: a base-piloto de 12 plantas e o conjunto de 80 consultas (com *fallback* $N = 11$, de baixo poder estatístico); e o roteador em produção, ainda léxico, sobre o qual a análise por limiar foi conduzida apenas *offline*. O sistema integra um projeto já aprovado por comitê de ética e será testado em uma unidade de saúde, sob a premissa de que o uso clínico pressupõe profissional habilitado.

Como trabalhos futuros, pretende-se: (i) ampliar a base curada e o conjunto de avaliação para aumentar o poder estatístico, sobretudo do caminho de *fallback*; (ii) refinar a rubrica e adotar um painel de múltiplos juízes (ou *grounding* obrigatório) para elevar a concordância com a especialista; (iii) substituir o roteador léxico por um roteador por limiar de similaridade, calibrado a partir da análise ROC, reduzindo o acionamento indevido do *fallback*; (iv) avaliar a reutilização do padrão em outro domínio de saúde, medindo o esforço de adaptação; e (v) comparar diferentes modelos de *embedding* e LLMs, investigando alternativas locais e abertas em conformidade com a LGPD.

Disponibilidade de Artefatos

O não compartilhamento dos artefatos de software, códigos-fonte, documentação técnica detalhada e demais informações relacionadas à implementação do sistema justifica-se pelo fato de o projeto encontrar-se atualmente em processo de registro de propriedade intelectual junto ao Instituto Nacional da Propriedade Industrial (INPI).

Agradecimentos

Declaração de uso de IA Generativa. Os autores utilizaram ferramentas de IA Generativa, incluindo *Gemini*, *ChatGPT* e *Claude*, para apoio em atividades de redação, revisão linguística, tradução e organização de trechos do manuscrito. O uso dessas ferramentas não exime os autores da responsabilidade integral pelo conteúdo, pelas análises e pelas conclusões apresentadas neste trabalho.

Referências

[1] Syed Adeel Ahmed, Gautam Gahlawat, Fahad Imam, and G. Saranya. 2024. AyurChat: Empowering Health through AI-Driven Ayurvedic Guidance. In *International Conference on Computing, Communication and Networking Technologies (ICCCNT)*. doi:10.1109/icccnt61001.2024.10724133

[2] Gisele Damian Antonio, Charles Dalcanale Tesser, and Rodrigo Otavio Moretti-Pires. 2014. Phytotherapy in primary health care. *Revista de Saúde Pública* 48, 3 (Jun 2014), 541–553. doi:10.1590/S0034-8910.2014048004985

[3] Sebastian Baltes, Florian Angermeir, Chetan Arora, Marvin Muñoz Barón, Chunyang Chen, Lukas Böhme, Fabio Calefato, Neil Ernst, Davide Falessi, Brian Fitzgerald, Davide Fucci, Junda He, Christoph Treude, Marcos Kalinowski, Stefano Lambiase, Daniel Russo, Mircea Lungu, Cristina Martinez Montes, Lutz Prechelt, Paul Ralph, Rijnard van Tonder, and Stefan Wagner. 2026. Guidelines for Empirical Studies in Software Engineering involving Large Language Models. *Empirical Software Engineering* (2026). arXiv:2508.15503 [cs.SE] https://arxiv.org/abs/2508.15503

[4] Victor R. Basili, Gianluigi Caldiera, and H. Dieter Rombach. 1994. The Goal Question Metric Approach. In *Encyclopedia of Software Engineering*, John J. Marciniak (Ed.). Vol. 1. John Wiley & Sons, 528–532.

[5] Brasil. 2009. *Política Nacional de Plantas Medicinais e Fitoterápicos.* Ministério da Saúde, Brasília. https://www.gov.br/saude/pt-br/composicao/sectics/plantas-medicinais-e-fitoterapicos/ppnpmf

[6] Brasil. 2010. *Programa Farmácia Viva: a integração das práticas tradicionais e complementares ao SUS.* Ministério da Saúde, Brasília. https://www.gov.br/saude/pt-br/composicao/sectics/plantas-medicinais-e-fitoterapicos/farmacia-viva

[7] Brasil. 2015. *Política Nacional de Práticas Integrativas e Complementares no SUS: 10 anos.* Ministério da Saúde, Brasília. https://bvsms.saude.gov.br/bvs/publicacoes/politica_nacional_praticas_integrativas_complementares_2ed.pdf

[8] Vitor Colombo, Weslen Souza, Lisane Brisolara, Brenda Santana, and Tatiana Tavares. 2025. Avaliação do uso de LLMs para Suporte à Tomada de Decisão na Adoção de Padrões de Projeto em Software Orientado a Objetos. In *Anais do XIX Simpósio Brasileiro de Componentes, Arquiteturas e Reutilização de Software* (Recife/PE). SBC, Porto Alegre, RS, Brasil, 101–111. doi:10.5753/sbcars.2025.14602

[9] Wenqi Fan, Yujuan Ding, Liangbo Ning, Shijie Wang, Hengyun Li, Dawei Yin, Tat-Seng Chua, and Qing Li. 2024. A Survey on RAG Meeting LLMs: Towards Retrieval-Augmented Large Language Models. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining* (Barcelona, Spain) *(KDD '24)*. Association for Computing Machinery, New York, NY, USA, 6491–6501. doi:10.1145/3637528.3671470

[10] Tom Fawcett. 2006. An Introduction to ROC Analysis. *Pattern Recognition Letters* 27, 8 (2006), 861–874. doi:10.1016/j.patrec.2005.10.010

[11] Cynthia Freire, Julielle Martins, Maria Paulino, Kelly Barbosa Silva Santos, Jessé Pavão, Thiago Rocha, Renata Santos, and Aldenir Santos. 2020. *Fitoterapia no Sus: Um Território para a Educação Popular em Saúde.* Atena Editora. 77–83 pages.

[12] Willian Freire, Murilo Boccardo, Daniel Nouchi, Aline Amaral, Silvia Vergilio, Thiago Ferreira, and Thelma Colanzi. 2024. Large Language Model-based suggestion of objective functions for search-based Product Line Architecture design. In *Anais do XVIII Simpósio Brasileiro de Componentes, Arquiteturas e Reutilização de Software* (Curitiba/PR). SBC, Porto Alegre, RS, Brasil, 21–30. doi:10.5753/sbcars.2024.3833

[13] Shivam Kumar and Shanu Kumar. 2025. Intelligent System for Medicinal Plant Identification and Personalized Health Recommendations. In *International Conference on Intelligent Computing and Information Systems (ICICI).* doi:10.1109/icici65870.2025.11069827

[14] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020. Retrieval-augmented generation for knowledge-intensive NLP tasks. In *Proceedings of the 34th International Conference on Neural Information Processing Systems* (Vancouver, BC, Canada) *(NIPS '20)*. Curran Associates Inc., Red Hook, NY, USA, Article 793, 16 pages.

[15] Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. 2008. *Introduction to Information Retrieval.* Cambridge University Press, New York, NY, USA. doi:10.1017/CBO9780511809071

[16] Quinn McNemar. 1947. Note on the Sampling Error of the Difference Between Correlated Proportions or Percentages. *Psychometrika* 12, 2 (1947), 153–157. doi:10.1007/BF02295996

[17] Jose Pereira, Danyllo Albuquerque, Mirko Perkusich, Guillermo Rodríguez, Jorge Díaz-Pace, Kyller Gorgônio, and Angelo Perkusich. 2025. Toward Generating Microservice Architectures from Textual Requirements with Large Language Models. In *Anais do XIX Simpósio Brasileiro de Componentes, Arquiteturas e Reutilização de Software* (Recife/PE). SBC, Porto Alegre, RS, Brasil, 79–89. doi:10.5753/sbcars.2025.14591

[18] Guilherme Pohlmann, Gabriel Fischer, Rodrigo Righi, Cristiano Costa, and Alex Roehrs. 2024. ADPS: Providing asynchronous PubSub communication in an adaptive and scalable way in Edge-Fog-Cloud architectures for healthcare data traffic. In *Anais do XVIII Simpósio Brasileiro de Componentes, Arquiteturas e Reutilização de Software* (Curitiba/PR). SBC, Porto Alegre, RS, Brasil, 1–10. doi:10.5753/sbcars.2024.3813

[19] Santiago Sabato, Guillermo Rodríguez, Claudia Marcos, Santiago Vidal, José Pereira, Danyllo Albuquerque, and Mirko Perkusich. 2025. Automated Clustering of Microservices Using Natural Language Processing and Clustering Algorithms. In *Anais do XIX Simpósio Brasileiro de Componentes, Arquiteturas e Reutilização de Software* (Recife/PE). SBC, Porto Alegre, RS, Brasil, 90–100. doi:10.5753/sbcars.2025.14592

[20] Amanda Galdino Costa Silva, Reginaldo dos Santos Pedroso, and Regina Helena Pires. 2025. Desafios e Estratégias para Inserção da Fitoterapia na Atenção Primária de Saúde: Revisão Sistemática. *Revista Contemporânea* 5, 7 (jul. 2025), e8711. doi:10.56083/RCV5N7-126

[21] Douglas Rodrigues Torres, Eduardo Dias Wermelinger, and Aldo Pacheco Ferreira. 2025. Aplicação da Inteligência Artificial na Atenção Primária à Saúde: revisão de escopo e avaliação crítica. *Saúde em Debate* 49, 145 (Apr 2025), e10070. doi:10.1590/2358-289820251451007OP

[22] An C. Tran, Thien Khai Tran, N. Nhut, and Nguyen Huu Van Long. 2023. Building a deep ontology-based herbal medicinal plant search system. *Applied Computing and Informatics* (2023). doi:10.1007/s41870-023-01250-6

[23] Maria Concepcion S. Vera and Thelma D. Palaoag. 2023. Implementation of a Smarter Herbal Medication Delivery System Employing an AI-Powered Chatbot. *International Journal of Advanced Computer Science and Applications (IJACSA)* 14, 3 (2023). doi:10.14569/ijacsa.2023.0140358

[24] Ellen M. Voorhees and Dawn M. Tice. 2000. The TREC-8 Question Answering Track Report. In *Proceedings of the Second International Conference on Language Resources and Evaluation (LREC'00)*, M. Gavrilidou, G. Carayannis, S. Markantonatou, S. Piperidis, and G. Stainhauer (Eds.). European Language Resources Association (ELRA), Athens, Greece, 77–82.

[25] Claes Wohlin, Per Runeson, Martin Höst, Magnus C. Ohlsson, Björn Regnell, and Anders Wesslén. 2012. *Experimentation in Software Engineering.* Springer, Berlin, Heidelberg. doi:10.1007/978-3-642-29044-2