
UNIVERSIDADE ESTADUAL DA PARAÍBA
CAMPUS I
PRÓ-REITORIA DE PÓS-GRADUAÇÃO E PESQUISA
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIAS E TECNOLOGIA EM SAÚDE
MESTRADO EM CIÊNCIAS E TECNOLOGIA EM SAÚDE

JORDAN FERREIRA NUNES

# USO DA LLM COMO FERRAMENTA PARA AUXILIAR NO PROCESSO DE REVISÃO SISTEMÁTICA

CAMPINA GRANDE/PB
2025

**JORDAN FERREIRA NUNES**

# USO DA LLM COMO FERRAMENTA PARA AUXILIAR NO PROCESSO DE REVISÃO SISTEMÁTICA

Dissertação apresentada ao Programa de Pós-Graduação em Ciência e Tecnologia em Saúde da Universidade Estadual da Paraíba, como requisito parcial à obtenção do título em Mestre em Ciência e Tecnologia em Saúde.

**Área de concentração**: Inteligência Artificial aplicada à saúde

**Orientador**: Prof. Dr. Frederico Moreira Bublitz

**CAMPINA GRANDE/PB**
**2025**

N972u Nunes, Jordan Ferreira.

Uso da LLM como ferramenta auxiliar no processo de revisão sistemática [manuscrito] / Jordan Ferreira Nunes. - 2025.

53 p.

Digitado. Dissertação (Mestrado Profissional em Ciência e Tecnologia em Saúde) - Universidade Estadual da Paraíba, Pró-Reitoria de Pós-Graduação e Pesquisa, 2025. "Orientação : Prof. Dr. Frederico Moreira Bublitz , Departamento de Computação - CCT. "

1. LLM. 2. NLP. 3. Systematic review. 4. Inteligência artificial na saúde. I. Título

21. ed. CDD 006.3

Elaborada por Bruno R. F. de Lima - CRB - 15/1021 BC/UEPB

**JORDAN FERREIRA NUNES**

# USO DA LLM COMO FERRAMENTA PARA AUXILIAR NO PROCESSO DE REVISÃO SISTEMÁTICA

Dissertação apresentada ao Programa de Pós-Graduação em Ciência e Tecnologia em Saúde da Universidade Estadual da Paraíba, como requisito parcial à obtenção do título em Mestre em Ciência e Tecnologia em Saúde.

**Área de concentração**: Inteligência Artificial aplicada à saúde.

Aprovado em: 20/06/2025

**BANCA EXAMINADORA:**

Prof. Dr. Frederico Moreira Bublitz
Universidade Estadual da Paraíba (UEPB)

Kátia Elizabete Galdino

Profa. Dra. Kátia Elizabete Galdino
Universidade Estadual da Paraíba (UEPB)

Ketinlly yasmyne N. Martins

Dra. Ketinlly Yasmyne Nascimento Martins
Coordenadora de Laboratório – NUTES/UEPB

## AGRADECIMENTOS

Inicialmente, agradeço ao meu orientador, Prof. Dr. Frederico Moreira Bublitz, pelo apoio, orientação e dedicação durante todo o período do mestrado. Sua experiência e incentivo foram fundamentais para a realização deste trabalho.

Agradeço, com imensa gratidão, à minha família, pelo suporte incondicional, paciência e encorajamento constante ao longo de toda esta jornada acadêmica.

Ao Núcleo de Tecnologias Estratégicas em Saúde (NUTES) da Universidade Estadual da Paraíba, pela oportunidade de integrar este programa de mestrado e pelo ambiente propício ao desenvolvimento científico e profissional.

Aos professores do programa, agradeço pelos ensinamentos, incentivo e contribuições valiosas à minha formação.

Por fim, agradeço a Deus, por me conceder forças, sabedoria e perseverança para concluir mais esta etapa da minha vida.

"Nós só podemos ver um pouco do futuro, mas o suficiente para perceber que ainda há muito a fazer." (Alan Turing)

## RESUMO

A revisão sistemática consiste em uma investigação científica acerca de um tema de pesquisa. Um dos maiores desafios ao iniciar uma revisão sistemática consiste na definição do protocolo, em especial, na escolha dos termos de busca e na seleção de artigos relevantes nas primeiras rodadas de investigação. Isso, pois, o tema ainda é possivelmente novo para o pesquisador e o seu conhecimento acerca do tema ainda precisa ser aprofundado. Nesse contexto, acredita-se que o uso dos LLMs (Large language models) que consistem de modelos NLP (Natural Language Processing) treinados com uma larga escala de dados textuais possa ajudar os pesquisadores nessa etapa da Revisão Sistemática (ou qualquer outra revisão de literatura). Mediante esse cenário, esse trabalho apresenta uma ferramenta desenvolvida com uso de LLM na forma de chatbot que desempenha um papel inicial de definir/encontrar sinônimos relevantes para os termos de busca, assim como identificar artigos relevantes com base nos critérios de busca. Atualmente a busca é realizada apenas na base de dados da Pubmed e para os sinônimos e artigos retornados o chat oferece a opção de explicar o porquê que tal artigo é considerado relevante. Um comparativo inicial dos termos relevantes e artigos retornados com um protocolo já definido mostrou que a ferramenta é capaz sim de ajudar o pesquisador, tendo levado um leigo a um termo de busca similar ao definido no protocolo em alguns minutos



**Palavras-chave**: llm; npl; systematic review.

**ABSTRACT**

A systematic review consists of a scientific investigation into a research topic. One of the biggest challenges when starting a systematic review is defining the protocol, especially choosing the search terms and selecting relevant articles in the first rounds of research. This is because the topic is possibly still new to the researcher and their knowledge of the subject has yet to be explored in greater depth. In this context, it is believed that the use of LLMs (Large language models), which consist of NLP (Natural Language Processing) models trained on a large scale of textual data, can help researchers at this stage of a Systematic Review (or any other literature review). Against this scenario, this work presents a tool developed using LLM in the form of a chatbot that plays an initial role in defining/finding relevant synonyms for the search terms, as well as identifying relevant articles based on the search criteria. Currently, the search is only carried out in the Pubmed database and for the synonyms and articles returned, the chatbot offers the option of explaining why that article is considered relevant. An initial comparison of the relevant terms and articles returned with an already defined protocol showed that the tool is indeed capable of helping the researcher, having led a layperson to a search term similar to the one defined in the protocol in a few minutes.

**Keywords**: llm; nlp; systematic review.

## LISTA DE ILUSTRAÇÕES

## LISTA DE TABELAS

## LISTA DE ABREVIATURAS E SIGLAS

| | |
|---|---|
| IA | *Inteligência Artificial* |
| CSV | *Comma-separated values* |
| DL | *Deep Learning* |
| GAN | *Redes Adversárias Generativas* |
| GPU | *Unidade de Processamento Gráfico* |
| LLM | *Large Language Model* |
| ML | *Machine Learning* |
| NLP | *Natural Language Processing* |
| RAG | *Retrieval-Augmented Generation* |
| RS | *Revisão Sistemática* |
| TPU | *Unidade de Processamento de Tensor* |

## LISTA DE QUADROS

**SUMÁRIO**

# 1 INTRODUÇÃO

A revisão sistemática consiste em uma investigação científica sobre uma pergunta formulada sobre um tema, onde com um protocolo bem ação bem definido busca nas literaturas acadêmicas disponíveis sobre o tema com o objetivo de selecionar as melhores evidências literárias disponíveis em relação ao tema. Uma revisão bem feita é considerada a melhor evidência para responder sobre determinado tema e possuir a característica de se outro pesquisador usa (TAíS *et al.*, 2014).

Um revisão sistemática possui etapas que necessitam de um extenso trabalho aplicado para ser desenvolvida como por exemplo há seleção da literatura muitas etapas manuais, como seleção de artigos e extração dos dados que geralmente fica na casas das centenas dependendo do tema pode ir para casa dos milhares tornando o trabalho árduo selecionar os artigos que mais se adequam ao tema pesquisado (MARTIN *et al.*, 2017).

> "Em geral um revisão sistemática é desenvolvida em grupo ou com ajuda de colaboradores muitas vezes o autor pede ajuda a colaboradores para dividir o trabalho com o objetivo de acelerar essas etapas mas mesmo assim pode demorar entre meses a semanas para serem concluídas" (MARTIN *et al.*, 2017).

Com o avanço das *LLMs*, acredita-se que grande parte desse esforço inicial possa ser simplificado, com uso de ferramentas de *NLP* para extrair informações relevantes de artigos. Assim, antes de iniciar o desenvolvimento do sistema, foi realizada uma revisão integrativa *(scopping review*) para responder a seguinte questão de pesquisa: *O método* NLP *pode ser/tem sido utilizado para extração de informação no contexto de RS?* Essa revisão demonstra que, apesar de relativamente nova e sem aplicações diretas ao contexto de RS, a hipótese tem grande potencial de ser válida.

Nesse sentido, foi desenvolvida neste trabalho uma *RAG* (*Retrieval-Augmented Generation*) integrado a uma *LLM* no formato de *Chatbot*, com o objetivo desenvolver uma aplicação com os *abstracts* vindos dos artigos providos pela busca na literatura, onde o pesquisador terá os artigos classificados por uma nota e uma explicação da atribuição da nota ao artigo levando como base relevância do artigo há *string* de busca utilizada. Inicialmente, as buscas estão sendo feitas apenas nas bases da Pubmed (tentou-se fazer uso de outras bases, mas para focar na viabilidade da pesquisa, a adição de novas bases de busca está no contexto de melhorias e trabalhos futuros). Apesar dessa limitação a ferramenta, que será descrita ao longo desse trabalho mostrou-se eficiente em definir palavras-chaves (a serem incluídas nos termos de pesquisa e na identificação de artigos

relevantes. Para cada um destes é possível ter uma explicação do porquê da inclusão deles.

# 2 OBJETIVOS

Com base nas abordagens identificadas na revisão de escopo descrita na sessão 4, observou-se que o método *Retrieval-Augmented Generation* (*RAG*) tem sido aplicado em diversos contextos, como assistente para diagnóstico de doenças infecciosas (STEWART *et al.*, 2024) e como chatbot no suporte à arquitetura de sistemas de emissão zero (GIHAN *et al.*, 2024). Inspirado por essas aplicações, o presente trabalho pretende explorar o uso de *Large Language Models* (*LLMs*) em conjunto com o método *RAG* para o desenvolvimento de um assistente voltado ao apoio de pesquisadores na condução de revisões sistemáticas da literatura.

Especificamente, este trabalho propõe o desenvolvimento de dois assistentes voltados para a etapa de construção da *string* de busca. O primeiro assistente é responsável por sugerir sinônimos para termos fornecidos pelo pesquisador, facilitando a ampliação e refinamento da *string* de busca. O segundo assistente realiza a análise dos *abstracts* dos artigos retornados pela *string* de busca na base de dados PubMed, atribuindo uma nota de relevância a cada artigo, acompanhada de uma justificativa textual que explica sua pertinência em relação à *string* informada. Dessa forma, busca-se tornar o processo de revisão sistemática mais eficiente, preciso e acessível.

# 3 FUNDAMENTAÇÃO TEÓRICA

## 3.1 REVISÃO SISTEMÁTICA

A revisão sistemática é definida como a aplicação de estratégia para limitar os vieses durante a montagem, utilizando uma avaliação crítica e sintetizando todos os estudos relevantes de um tópico específico (IAIN *et al.*, 2002). Atualmente considerada padrão ouro em comparação aos demais tipos de revisão literária, possui uma longa história começando no século XVIII pelo Dr. James Lind, cirurgião naval escocês, que realizou estudos sobre a natureza, causa, prevenção e tratamento da doença escorbuto (JEAN *et al.*, 2023). (IAIN *et al.*, 2002) rastreou algumas revisões públicas no início do século XX em várias áreas como medicina, agricultura, física e outros. Inúmeras artigos usaram ou advogaram métodos sistemáticos para revisar a literatura podem ser achados na primeira metade do século XX, passando a ter mais força em 1970, quando a síntese de pesquisa começou a ter destaque necessitando aplicar melhores transparência e melhores métodos para validar as revisões.

### 3.1.1 Metodologia de uma RS

Uma das orientações recomendadas para uso na área da saúde é são as orientações providas pelo método Prisma, para auxiliar no desenvolvimento da revisão, de modo que caso seguia as orientações descritas no apêndice A, há uma garantia que sua revisão tenha todos as informações necessárias, elaboração da palavra-chave da pesquisa, seleção das publicações, busca de artigos, extração dos dados desses artigos, sínteses dos dados, análise de relevância dos artigos e conclusão, para uma revisão.

### 3.1.2 Metanálise

É um método estatístico para agregação de resultados abundantes de pesquisas independentes, onde se busca a sinterização dos resultados obtidos usando a estatística como esse propósito. Podendo obter, por meio dessa síntese, pode obter novos resultados de pesquisa, comparar resultados contraditórios e expandir teoria ou propor novas (MARCOS *et al.*, 2009).

#### 3.1.2.1 A escolha dos dados para aplicar a metanálise

Escolher quais resultados dos estudos serão aplicados à metanálise, em saúde, é comum o uso da tecnologia (medicamentos, técnicas de tratamento, procedimentos, etc.) de que se trata o estudo para aplicação da metanálise.

#### 3.1.2.2 A escolha entre os modelos de efeito fixo e modelo de efeito aleatórios

O modelo de efeitos fixos parte do princípio de que os efeitos das intervenções seguem a mesma linha, tanto em magnitude quanto em direção, em todos os estudos que compõem aquela metanálise. É como se pensássemos que todas as variáveis dos estudos (aplicação da intervenção, população, método de mascaramento. . . ) são semelhantes, ou até mesmo idênticas, e que as possíveis diferenças observadas entre os resultados se devem única e exclusivamente ao acaso ou ao erro amostral. Assim, as heterogeneidades dos estudos são ignoradas. No modelo de efeito aleatórios, presume-se que os resultados dos estudos incluídos na metanálise seguem uma distribuição normal (aquela da curva de Gauss ou curva em forma de sino) e que as diferenças observadas nesses resultados se devem não somente ao acaso, mas também a diferenças existentes entre os próprios estudos, como a população, por exemplo (diversidade ou heterogeneidade clínica e metodológica). Embora a escolha em geral seja por modelos aleatórios, tem que ter cuidado para a amostra dos artigos for muito pequena, os cálculos probabilísticos podem ser pouco acurados.

### 3.1.3 Análise estatística

#### 3.1.3.1 Teste Q de Cochran

É um teste estatístico não paramétrico para medir se os estudos variam entre estudos, determinando se eles são parecidos entre si ou não, onde a hipótese nula é que os estudos são homogêneos.

A fórmula do Teste Q de Cochran é:

$$Q = \frac{(k-1)\sum_{i=1}^{k}(T_i - \bar{T})^2}{\sum_{i=1}^{k} R_i - \frac{1}{k}\sum_{i=1}^{k} R_i^2}$$

- $Q$ é o valor do teste Q de Cochran
- $k$ é o número de grupos

- $T_i$ é o total de sucessos no grupo $i$
- $\bar{T}$ é a média dos totais de sucessos em todos os grupos
- $R_i$ é o total de sucessos em cada bloco (sujeito)

3.1.3.2 Estatística I²

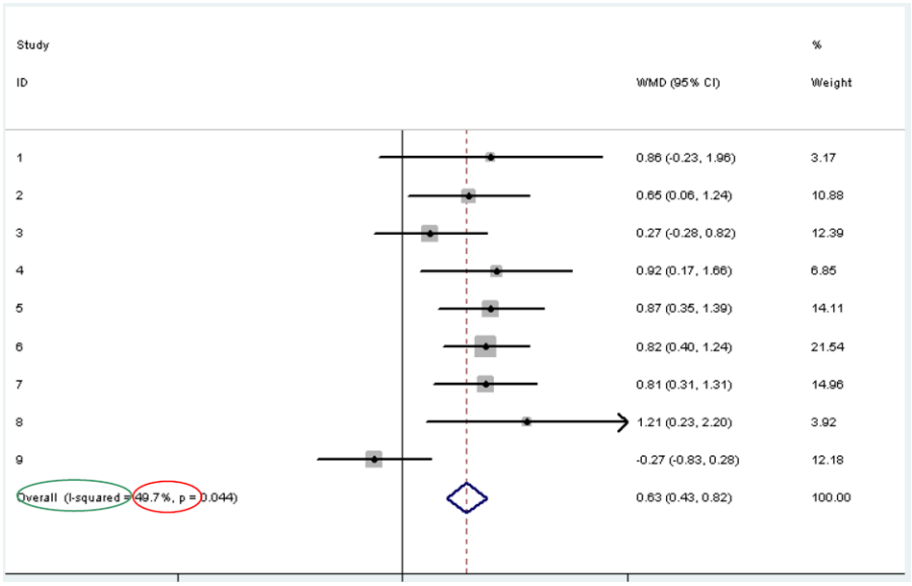
Outro teste estatístico para medir a heterogeneidade dos estudos que compõem a metanálise é o parâmetro Q, obtido usando o método Q de Cochran, J é a quantidade de estudos contidos na metanálise. Quanto mais perto de 0, mais homogêneos são os estudos.

$$I = \frac{Q - (J - 1)}{Q}$$

3.1.3.3 Forest plot

É a forma comumente usada para apresentar o resultado da metanálise. O gráfico presente na Figura 3 mostra o intervalo de confiança para cada estudo presente na metanálise. Os símbolos presentes nos gráficos são proporcionais à relevância do estudo para a metanálise.

**Figura 1 –** *Florest Plot*




**Fonte**: Elaborada pelo autor, 2025.

## 3.2 PROCESSAMENTO DE LINGUAGEM NATURAL

NLP, sigla para processamento de linguagem natural, é um dos ramos da inteligência artificial(IA) com foco em como o computador pode processar a linguagem humana. Atualmente, ferramentas de *NLP* como o *Chat GPT* da *openIA*, *Llama* do Meta dentro de outras *LLMs* combinam IA mais estatísticas para poderem predizer a próxima palavra em uma sentença baseada nas palavras processadas anteriormente. Modelos de *NLP* também são usados para tarefas mais simples, como classificação de documentos, análise de sentimento de um bloco de texto, e para tarefas mais complexas, como responder perguntas, sumarização de relatórios, dentre outros.

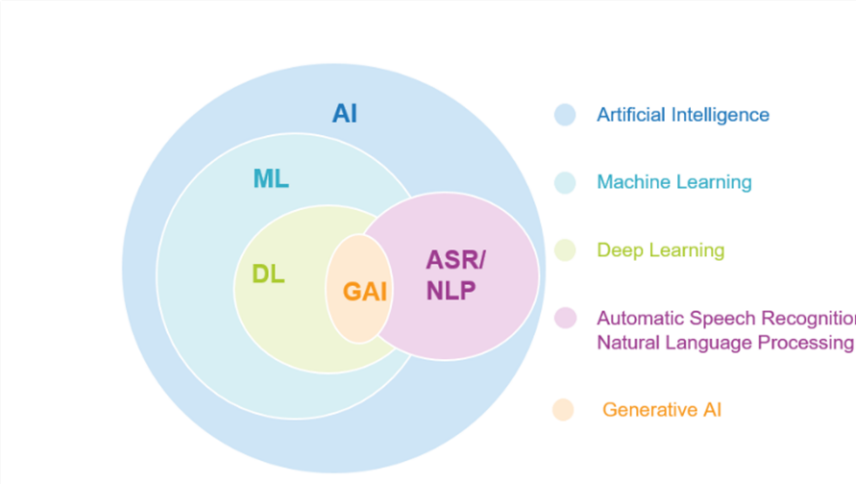
### 3.2.1 Embbedings

Os *embbedings* em *NLP* são modelos no qual a função é capacitar o computador a processar, compreender e gerar textos, são modelos cuja função é transformar dados em linguagem natural para uma matriz vetorial, mantendo sua semântica e significado(YOSHUA *et al.*, 2003).

### 3.2.2 IA Generativa

*IA Generativa* tem uma história curta, sendo introduzida no meio durante 1960 com o início do desenvolvimento de *Chatbot*. Atualmente é uma inteligência artificial capaz de gerar imagem, vídeo, áudio, texto em segundos, mas isso só foi começando a virar realidade em 2014 com o conceito de *generative adversarial network* (GAN), é uma rede neural profundo capaz de gerar novos dados autênticos a partir do treinamento de 2 redes neurais que competiam entre si utilizando um base de dados determinada, a partir deste ponto começou a evoluir (KEITH; D.; FOOTE, 2024).

**Figura 2 –** *Diagrama como a IA generativa está alocada no campo da IA (2023)*




**Fonte**: Cheonsu e Jeong (2023).

É uma forma de inteligência artificial que usa uns grandes volumes de dados de treinamento para gerar um novo contexto, podendo ser texto, imagem, áudio ou vídeo. Exemplos de *Generative IA* é o *Llama 3.2* o qual é um modelo de linguagem treinado por uma vasta base de dados e *Dall-e* focado na geração de imagens. A categorização dos modelos de *IA generativa* baseia-se no tipo de saída gerada, com a distinção entre modelos de linguagem, de imagem, vídeo, entre outros. Entretanto, o horizonte se desenvolveu rapidamente para modelos que podem aprender tanto por imagem ou texto simultaneamente. No contexto de inteligência artificial, *IA generative* abrange tanto modelos não supervisionados, modelos que utilizam como dado de entradas bases de dados sem rótulo, como semi-supervisionados, modelos que utilizam dados semi-rotulados onde uma parte possui e outra não. Diferente dos modelos tradicionais de aprendizado de máquina que só analisam e processam dados já existentes, IA generativa aplica um novo conceito com a geração de novos dados e de conteúdo inédito (CHEONSU; JEONG, 2023).

**Figura 3 –** *Tipos de entradas de dados em Modelos*


**Fonte**: Marcus e Almeida (2023).

### 3.2.3 Llm

Os modelos de *LLM* (Large language model) são modelos de *NLP* treinados para serem de IA generativa, modelos que são capazes de gerar novos conteúdos nesse caso de texto, cujo são treinados com uma larga escala de dados textuais provenientes de diversas fontes, como site, pdfs livro dentre outras, sendo uma poderosa ferramenta para confecção de tradutores, *Chatbot*, geração de texto e dentre outras funções.

Os modelos caracterizam-se por terem uma alta capacidade de geração de novos dados e de processar grandes volumes de dados. Entretanto, para mantê-las e acrescentar novo conhecimento não é uma tarefa fácil, pois necessitam de imenso volume de dados para treinamentos além de uma infraestrutura de *GPU* OU *TCU* para retreina-la mesmo usando técnicas avançadas de transferência de conhecimento qLora essa etapa é demorada e custosa.

O primeiro passo é treinar como é a coleta de dados de diversas fontes de dados, como livros, websites, artigos, dataset públicos, dentre outros. Desenvolver mais proficientes usam-se fontes que contenham bases pré-treinadas de texto, as quais são do mesmo tipo como base de conhecimento geral ou específico. As bases de conhecimento gerais são encontradas em livros, websites, conversas de texto e visam melhorar a generalização da *LLM*. As bases específicas, tais como código, estudos científicos, textos em múltiplas linguagens, pretendem melhorar a solução de tarefas específicas da *LLM* (CHEONSU; JEONG, 2023).

O próximo passo é limpeza dos dados adquiridos, removendo de sentenças réplicas, alguns filtros de linguagem para retirar pontuação, palavras indesejáveis como as *stopword*, tais como artigos, preposições, conjunções e outras, remoção de Substantivo próprio, realizar uma tokenização nas sentenças e por fim aplicar um algoritmo de *embeddings*.

O terceiro passo é a configuração, seleção de números de camada da rede, do *attencion head*, método que simula a atenção humana para definir pesos das palavras contidas na rede, definição da função de perda e dos demais hiper parâmetros da rede.

Treinamento do modelo, nessa parte necessita uma robusta infraestrutura para rede, pois além de conter milhões a bilhões de parâmetros são treinados bilhões ou trilhões de entradas, esse processo se repete milhares de vezes para atualizar os pesos contidos na rede tornando capaz de prever a próxima palavra de uma sentença.

O refinamento e avaliação é o último passo, onde se passa como entrada um *dataset* de teste não usado durante a etapa de treinamento para verificar o desempenho da rede. Caso não esteja aceitável, realizar uma etapa de refinamento onde serão ajustados os hiper parâmetros da rede ou realizar um treinamento com dados adicionais para melhorar o desempenho (CHEONSU; JEONG, 2023).

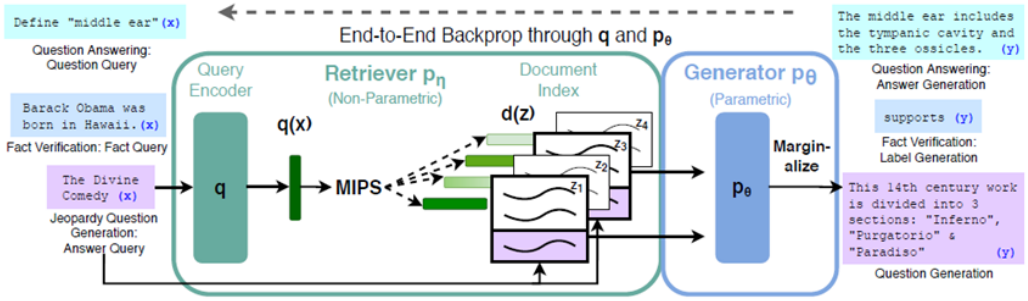
*QLora e Lora* são algoritmo para realizar refinamentos em modelos, utilizando quantização, diminuir os representatividade do ponto flutuante para diminuir o seu espaço em memória, dessa forma reduzindo o tamanho necessário de *GPU* para refinamento de modelos, podendo refinar modelos de 65 bilhões de parâmetros em uma *GPU* 48GB preservando sua representatividade nos pontos flutuantes (TIM *et al.*, 2023).

### 3.2.4 Rag

Um dos grandes problemas das *LLM* é a limitação de conhecimentos, pois as mesmas só possuem conhecimento até o período em que foi realizado o seu treinamento, fazendo conhecimentos posteriores à data de treinamento para não serem contemplados. Apesar desse problema, pode ser resolvido por meio de refinamento do treinamento, treinando-a com dados atuais, mas como se trata de modelos gigantescos com mais de bilhões de parâmetros, seu treinamento é demorado e custoso, pois precisa de arquitetura computacional pesada para realizar esse feito.

Uma forma para resolver esse problema foi a criação da *RAG* (*retrieval-augmented generation*), o qual é um modelo híbrido onde combinasse já pré-treinadas, com um recuperador de informação com o objetivo desse recuperar retornar N dados novos para a, onde a mesma selecionar quais desses dados seriam mais relevantes para constituir a resposta de uma determinada pergunta, mesmo a não possuindo conhecimento prévio do assunto abordado.

**Figura 4 –** *Arquitetura de uma RAG*




**Fonte**: atrick *et al.* (2021).

Na imagem acima demonstra-se um exemplo de uma *RAG*. Nesse contexto, quando uma pergunta ou argumento é passado como entrada (*Query Encoder*), o recuperador (*Retriever*) busca qual dos documentos pode responder essa pergunta, por meio de uma base de conhecimento própria. Para isso, ele usa o algoritmo de busca *MIPS* (*Maximum inner-product search*) para retornar os N documentos mais relevantes para responder à pergunta feita (*Document Index*). Por fim, esses documentos são usados pelo gerador (*Parametric Generator*) para formular uma resposta.

# 4 TRABALHOS RELACIONADOS

O principal objetivo neste capítulo é responder à seguinte questão de pesquisa: O *método NLP pode ser/tem sido utilizado para extração de informação no contexto de RS?* Para isso, foi utilizada uma abordagem de revisão da literatura utilizando o método de *scoping review*. Pois, nesse momento da pesquisa, é importante entender o tema de forma ampla, trazendo elementos que possam ampliar os conhecimentos acerca do tema. Isso também irá contribuir para estabelecer o nível de originalidade da solução proposta neste trabalho, uma vez que o método *RAG* é recente. Durante a pesquisa nas bases de dados pubmed e IEEE não foram encontrados artigos onde o método *RAG* está sendo utilizado diretamente no contexto *RS*, mas foram encontrados artigos onde está sendo utilizando para nos mais diversos contextos, seja no desenvolvimento de um assistente para vítimas de assédio sexual (SONIA *et al.*, 2024) ou desenvolvimento de uma interface de bate-papo sobre doenças hepáticas (JIN *et al.*, 2024).

## 4.1 STRING DE BUSCA (IEEE)

A String de busca *RAG* retorna 1365 resultados, isso se deve ao termo por ser um assunto mais abrangente e NLP ser só um subtópico deles, resultando em artigos que não têm relação com o assunto *RAG*. A String de busca composta somente a abreviação *RAG* tem 339 resultados, dentro deles tem outras abreviações como RAGged, RAGs que são campos de pesquisa totalmente diferentes do *RAG*, necessitando refinamento na string de busca, visando o foca da pesquisa que é utilizando da RAG para auxiliar a produção revisão sistemática foram utilizadas as seguintes strings de busca *RAG and RS* e *systematic review and RAG*, mas tiveram 0 resultados. Visando artigos com soluções similares da proposta, foram selecionados 2 artigos para um compor um grupo relevante são eles (SONIA *et al.*, 2024) e (GIHAN *et al.*, 2024) ambos têm ideias similares a proposta, com isso foram extraídas 2 palavras-chave desses artigos que foi *Chatbot* e *informartion retrieval*, tornado a String de busca *"RAG or and ("Information Retrieval" or "Chatbot")"* retornando 26 artigos no total.

## 4.2 CRITÉRIOS DE INCLUSÃO/EXCLUSÃO

O tema a ser estudado é relativamente novo, dado que a tecnologia *RAG* foi divulgada em meados de 2021, (ATRICK *et al.*, 2021). Sendo assim, não será utilizado nenhum limitador temporal. Note também que não há limitação em relação ao escopo em que a *RAG* foi utilizada, dado que a *string* de busca não retornou nenhum artigo

focado no tema específico de RS. Assim, foram definidos os seguintes critérios de inclusão e exclusão.

### 4.2.1 Inclusão

O artigo deve ter usado o método *RAG* visando extração de informação de uma base externa. A utilizada não foi treinada com esses dados.

### 4.2.2 Exclusão

Os artigos duplicados, o artigo com difícil acesso, não pode ser focado em melhoramento do método *RAG* ou com foco em formas para validar a *RAG*.

## 4.3 CRITÉRIOS DE EXTRAÇÃO

O objetivo nesta seção é estabelecer que tipo de informação será extraída dos artigos incluídos na revisão. Visando obter respostas acerca da utilização da *RAG* como método de extração em RS, estabeleceram-se os seguintes critérios:

- Base de dados: no contexto das *RAG* uma base de dados consiste em variadas fontes de informação, podendo ser pdf, texto, banco de dados, etc. É importante entender como a base de dados foi "criada" e as características dessa base de dados.
- O assunto principal das bases de dados: Cada base de dados está relacionada a um domínio de conhecimento específico (ou assunto). No contexto deste trabalho, é importante saber qual o assunto utilizado na base de dados para operar, pois por não ter um critério de base específico e podendo operar em várias bases diferentes, bastando ela ser uma base de dados do tipo vetorial.
- Resultados: Verifica o resultado obtido pelo autor, o método RAG verifica se o resultado alcançado foi satisfatório ou não para o contexto aplicado.

### 4.3.1 LLM Utilizada

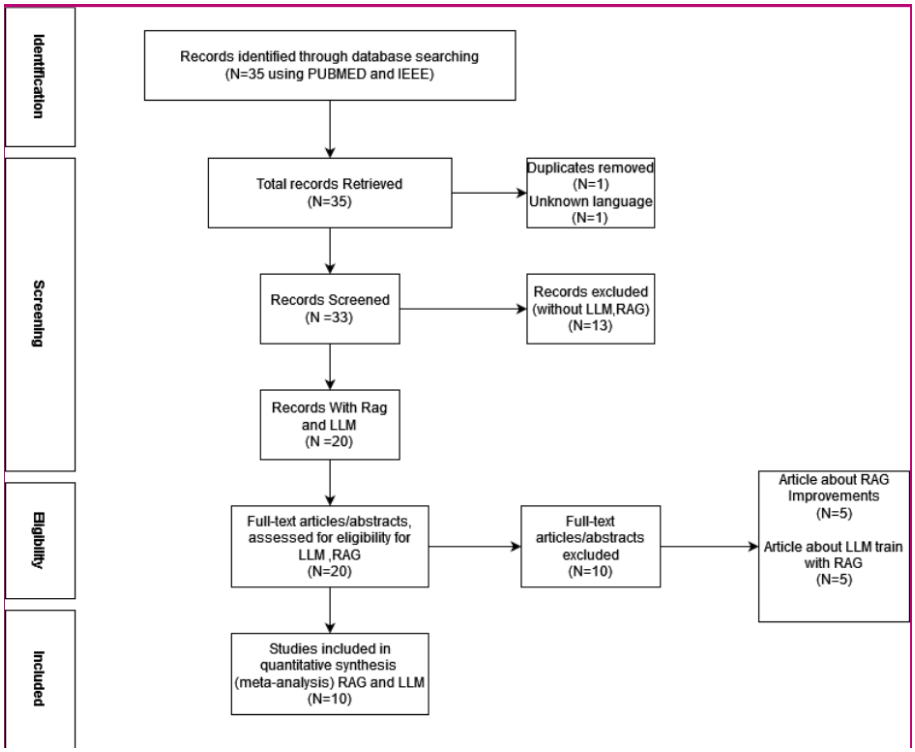
- Llama2: Desenvolvida pela empresa Meta lançada em 2023 e sucessor do modelo Llama 1 de 2022 foi treinado com mais de 2 trilhões de token e aceita contexto de 4096 token, contendo modelos de 7B,13B e 70B de pesos, é disponível gratuitamente para uso comercial e em pesquisa.

- Chatgpt: Desenvolvida pela openIA de uso restrito e necessita de licença para ser usada.
- Mistral: LLM de código aberto, criada em 2023 para concorrer com o ChatGPT da openIA.

### 4.3.2 Embedded

- Ada Text Embedding version 2:*Embedding* adquirido da empresa *OpenIA*, com uma saída de um vetor com 1596 de dimensão, contendo uma rica captura de semânticas das frases transformadas por ele.
- Med-BERT: *Embedding* treinado com o modelo *Bert*, treinado com prontuários eletrônicos de pacientes para ser mais assertivo para a transformação de texto em vetor, no contexto médico.
- Bert: Embedding criado pela Google, treinado de forma não supervisionada, usado uma abordagem bi direcional em seu treinamento onde a frase João tem uma bela casa, a palavras contidas das frases terão representantes tanto a direta quanto a esquerda, esse modelo gerou um novo estado da arte para os embedding.
- Sentence-BERT, msmarco-bert-base-dot-v5: Modelos que foram treinados a partir do modelo de *embbeding Bert*.

**Figura 5 –** Revisão de escopo




**Fonte**: Elaborada pelo autor, 2025.

Nos artigos serão extraídos o assunto principal das bases de dados, a utilizada, o algoritmo de embedding usado, o resultado alcançado com o método *RAG* e caso tenha as melhorias feitas no método.

**Tabela 1 –** Extração de dados dos artigos - Parte 1

| Título | Autor | Local | Ano | Modelo LLM | Modelo Embeddings | Assunto | Base de dados | Resultados |
|---|---|---|---|---|---|---|---|---|
| Development of a Liver Disease-Specific Large Language Model Chat Interface using Retrieval Augmented Generation | Jin Ge , Steve Sun , Joseph Owens | San Francisco | 2023 | Chatgpt-4 | Ada Text Embeddings version 2 | Doenças relacionada ao fígado | Documentação pública do AASLD practice guidelines | Foi alcançado um bom resultado, mas uma das melhorias sugeridas foi a adição de mais dados para evitar viés de contexto, isso no futuro, 3 de 10 das perguntas foram respondidas de maneira incompleta. |
| Overcoming LLM Challenges using RAG-Driven Precision in Coffee Leaf Disease Remediation | Dr. Selva Kumar S | Bangalore, India | 2024 | Chatgpt-3.5 | Open IA Embeddings | Diagnóstico para doenças em folhas de café | Kaggle, open-source database e dados reais de doença Coffee Leaf | A utilização da técnica RAG minimizou o efeito de alucinação das LLM, onde ela produz resultados inconsistentes, melhorando os resultados. |
| Evaluating and Enhancing Large Language Models' Performance in Domain-specific Medicine: Explainable LLM with DocOA | Xi Chen, Li Wang | China | 2024 | DocOA, Chatgpt-3.5, Chatgpt-4 | Med-BERT | Osteoartrite | Diretrizes médicas para Ortopédica e ficha de 80 pacientes com osteoartrite. | A utilização do método Rag melhorou tanto a performance quanto a explicabilidade , onde previamente só era usado a LLM normal. |
| Unlocking Telecom Domain knowledge using LLMs | Sujoy Roychowdhury | | 2024 | llama2-7b-chat model | Sentence-BERT | Base de conhecimento da Telecom | Especificações 3GPP de domínio público, especificações de 802.11 e 3GPP 38.101-1 contidas nos sites 3GPP ou ETSI sites e IEEE | O resultado não foi satisfatório pelos motivos: o parser utilizado para os documentos em formato de Latex não eram precisos acarretando em perda de informações além dos documentos conter tabelas com grande dimensões tornando o parser impreciso perdendo informação no processo |
| RAG-based LLM Chatbot using Llama-2s | Sonia Vakayil | India | 2024 | llama2-7b-chat model | msmarco-bert-base-dot-v5 | Asistencia a vitimas de assédio sexual. | PDFs que contenha informações sobre leis e recursos relativos a assédio sexual. | A llm obteve um bom resultado pois gerou orientação as vitimas de assédio sexual, as respostas não julgava as vitimas e contia empatia, provento o suporte almejado, só ocorrendo no caso de assédio sexual aos homem mas isso segundo o autor foi devido a pouca informação e relato sobre esse crime. |

**Fonte**: Elaborada pelo autor, 2025.

**Tabela 2 –** Extração de dados dos artigos - Parte 2

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Multi-Agent RAG Chatbot Architecture for Decision Support in Net-Zero Emission Energy Systems | Gihan Gamage | | 2024 | chatGPT-4 | text-embedding-ada-002 | As empresas querem aderir ao sistema zero de emissão. | Diversas fontes de dados tanto estruturados (banco de dados) como não estruturados (pdf, texto) sobre sistema de zero de emissão | O chatbot foi integrado com sucesso e avaliado empiricamente com casos de uso do mundo real para ajudar tomada de decisão no sistema energético com emissões líquidas zero de uma grande instituição de ensino superior. |
| A RAG-based Medical Assistant Especially for Infectious Diseases | Stewart Kirubakaran S | India | 2024 | Mistral-7B | Open IA Embeddings | Covid-19 | Pdf e text da empresa Elsevier do new york times. | A Rag conseguiu gerar resposta bem informada além de utilizar dados atualizados sobre covid. |
| A General Approach to Website Question Answering with Large Language Models | Yilang Ding | USA | 2024 | GPT-3.5-turbo | all-MiniLM-L6-v2 | Informaçõe s que estão disponíveis na Emory University's. | Um conjunto de sites escolhidos para o propósito do paper. | Apresentou um bom resultado nas perguntas mais simples do teste. |
| :Enhancing Drug Safety Documentation Search Capabilities with Large Language Models: A User-Centric Approach | Jeffery L. Painter | Londres | 2023 | chatGPT-4 | text-embedding-ada-002 model. | Busca por documento s complexos no contexto de Farmacia | Uso do dados contidos no Global Safety Database | A Rag apresentou alucinações, principalmente quando o dado era do tipo tabular, mas mostrou que é capaz de recuperar informações de documentos complexos precisando de ajuste para sua melhor adaptação. |
| Addressing the Productivity Paradox in Healthcare with Retrieval Augmented Generative AI Chatbots | Sajani Ranasinghe | USA | 2024 | GPT-3.5 | Bert | Uso de chatbot nos desafios aplicados nos sistemas de saúde complexos | Prontuário eletrônico, conversas com pacientes, publicações , teste científicos na área de saúde | A Rag apresentou apresentou potencial no processamento de informação e tomada de decisão para um serviço de saúde |

**Fonte**: Elaborada pelo autor, 2025.

# 5 ETAPAS METODOLÓGICAS

Essa pesquisa possui uma abordagem qualitativa, com aspectos descritivos, fundamento positivista e de natureza aplicada (prática). Foi concluído no ano de 2025 para a dissertação do programa de Mestrado Profissional em Ciência e Tecnologia em Saúde, oferecido pelo Departamento de Computação da Universidade Estadual da Paraíba (UEPB), Campus 1, Campina Grande - Paraíba.

## 5.1 PROCEDIMENTOS REALIZADOS

Para a escrita desse trabalho realizou-se uma revisão de escopo com os termos de RAG, *chatbot* e *Information Retrieval* nas bases de artigos e publicações *PubMed* e *IEEE Explorer*. Foi considerado artigos, revistas, jornais, monografias, dissertações, entre outros tipos de pesquisas, somente no idioma inglês e não houve definição de período. Houve definição de critérios de inclusão, exclusão,aceitação e extarção como descrito na Seção 4.2.

O desenvolvimento da aplicação seguiu uma abordagem iterativa, iniciando pela definição das fontes de dados bibliográficas e das tecnologias a serem utilizadas para a construção dos assistentes baseados em modelos de linguagem. Inicialmente, foram investigadas quais bases de dados científicas possuíam bibliotecas compatíveis com a linguagem de programação *Python*, essencial para a integração com a aplicação. As plataformas *Scopus* e *PubMed* foram identificadas como alternativas viáveis. Contudo, a biblioteca da Scopus requer uma conta institucional vinculada ou uma assinatura paga, o que inviabilizou seu uso neste projeto. Dessa forma, optou-se pela utilização da base de dados PubMed, que permite acesso gratuito mediante a solicitação de uma chave de *API*.

Para o modelo de LLM, foi escolhida a família *LLaMA (Large Language Model Meta AI)*, por apresentar uma licença de uso aberta, tanto para fins acadêmicos quanto comerciais. A LLM foi obtida através do repositório *Hugging Face*, utilizando-se a biblioteca *Transformers* para seu carregamento e manipulação. Essa biblioteca foi empregada tanto no desenvolvimento dos assistentes quanto na implementação da arquitetura *RAG (Retrieval-Augmented Generation)*, permitindo a integração entre recuperação de documentos e geração de linguagem natural.

Inicialmente, os protótipos foram construídos utilizando o ambiente de desenvolvimento *Jupyter Notebook*, o que facilitou a experimentação e os testes das funcionalidades básicas dos assistentes. Posteriormente, com o objetivo de oferecer uma interface mais

amigável e acessível aos usuários, foi desenvolvida uma aplicação web utilizando a biblioteca *Streamlit*, que proporciona uma construção rápida e interativa de interfaces gráficas. O ambiente de desenvolvimento utilizado para esta etapa foi o *Spyder*, o qual ofereceu suporte para organização e manutenção do código-fonte.

Essa abordagem permitiu não apenas a validação funcional dos assistentes em um ambiente controlado, mas também a elaboração de uma interface que facilita o uso da aplicação por parte de usuários com diferentes níveis de familiaridade técnica

## 5.2 FERRAMENTAS UTILIZADAS

O desenvolvimento deste trabalho foi realizado utilizando um conjunto de ferramentas e bibliotecas fundamentadas na linguagem de programação *Python*, escolhida por sua ampla adoção na área de ciência de dados, inteligência artificial e desenvolvimento web.

Para a construção da interface visual da aplicação web, foi utilizada a biblioteca *Streamlit*, que possibilita a criação rápida e intuitiva de interfaces gráficas interativas. Essa escolha proporcionou maior usabilidade ao sistema, permitindo que os assistentes fossem acessados de maneira prática por usuários com diferentes níveis de familiaridade técnica.

O núcleo do sistema foi implementado utilizando o modelo de linguagem *LLaMA 3.2*, integrado à biblioteca *Transformers*, responsável pelo carregamento e gerenciamento da *LLM*. Essa combinação foi essencial para a construção da arquitetura *RAG (Retrieval-Augmented Generation)*, que permite combinar a recuperação de informações relevantes com a geração de respostas contextuais.

A etapa de extração de dados científicos foi realizada por meio da biblioteca *Metapub*, que possibilita o acesso programático à base de dados *PubMed*, facilitando a recuperação de artigos científicos de maneira automatizada e eficiente.

Durante o processo de desenvolvimento e testes, foram utilizadas as IDEs *Jupyter Notebook* e *Spyder*, que permitiram a organização modular do código, a prototipação de funcionalidades e a validação progressiva do sistema.

Todas as ferramentas e bibliotecas utilizadas neste projeto têm como linguagem base o *Python*, o que favoreceu a integração entre os diferentes componentes do sistema e garantiu maior coesão na implementação da solução proposta.

# 6 SOLUÇÃO PROPOSTA

Neste capítulo está apresentada a aplicação proposta a qual é composta de 2 elementos o primeiro é um assistente de sinônimos, cujo objetivo é informar ao usuário uma tabela contendo 10 sinônimos a palavra pesquisada pelo usuário, para expandir a *string* de busca com os termos equivalentes ao consultado cujo possivelmente o usuário desconhecia enriquecendo a sua *string* de busca na fase de seleção de artigos na revisão sistemática. O segundo é um assistente de análise, cujo objetivo é mensurar a relevância do artigo encontrado pela *string* de busca com uma nota variando entre os intervalos de 0 a 1, onde 0 indica que artigo encontrado não possui relevância para revisão sistemática do usuário e 1 indica que possui forte relevância para a revisão do usuário, essa nota é feita com base nos termos de pesquisa da *string* de busca, onde é verificado se eles estão presente no *abstract* do artigo e se tem correlação ao quer o autor está pesquisado para explicar nota recebida pelo artigo é gerado um texto relatando os motivos para os quais a nota foi atribuída para o artigo em si com base na *string* de busca do usuário e o *abstract* do artigo, gerando um *rank* ordenado de maneira decrescente a nota atribuída, para auxiliar o usuário na escolha dos artigos relevantes para sua revisão sistemática.

## 6.1 COMPONENTES DA SOLUÇÃO

### 6.1.1 Llama 3.2

O *Llama 3.2* é a versão com os primeiros modelos multimodais da série do *LLama 3*. O *Llama 3.2* se concentra em duas áreas principais, *LLMs* habilitados para visão. Os modelos multimodais de parâmetros 11B e 90B agora podem processar e compreender textos e imagens e *LLMs* leves para borda e dispositivos móveis. Os modelos de parâmetros 1B e 3B foram projetados para serem leves e eficientes, permitindo que sejam executados localmente em dispositivos de borda. Na solução, foram utilizados os modelos *Llmma 3.2* com os parâmetros 1B e 3B, pois são modelos de menor porte que necessitam de menor poder computacional para rodar, diminuindo o custo de manter a solução. Os modelos com 1B de parâmetros foram utilizados no assistente sinônimo e os 3B no assistente de análise.

### 6.1.2 Transforms Hugging Face

Biblioteca visando estruturar de código aberto para aprendizagem profunda criada pela *Hugging Face*. Ela fornece APIs e ferramentas para baixar modelos pré-

treinados de última geração e ajustá-los ainda mais para maximizar o desempenho. Esses modelos dão suporte a tarefas comuns em diferentes modalidades, como processamento de linguagem natural, pesquisa visual computacional, áudio e aplicativos multimodais. Utilizada na solução para carregar e criar da *RAG*.

### 6.1.3 Rag

Processo de otimizar a saída de um grande modelo de linguagem, de forma que ele faça referência a uma base de conhecimento confiável fora das suas fontes de dados de treinamento antes de gerar uma resposta. Grandes modelos de linguagem (*LLMs*) são treinados em grandes volumes de dados e usam bilhões de parâmetros para gerar resultados originais para tarefas como responder a perguntas, traduzir idiomas e concluir frases. A RAG estende os já poderosos recursos dos LLMs para domínios específicos ou para a base de conhecimento interna de uma organização sem a necessidade de treinar novamente o modelo. Utilizado para referenciar o conhecimento obtido com a *Pubmed* para a *LLM* no assistente de análise.

### 6.1.4 Metapub

*Metapub* é uma biblioteca *Python* que fornece objetos *Python* recuperados via eutils (é um pacote Python que facilita o uso da interface de utilidades do *National Center for Biotechnology Information (NCBI)*) que representam artigos e conceitos do *PubMed* encontrados nas bases de dados do *NCBI* (*National Center for Biotechnology Information*). Foi utilizada na solução no assistente de análise para buscar artigos referentes à string de busca passada pelo usuário.

### 6.1.5 Streamlit

*Streamlit* é uma biblioteca open-source em *Python* que permite a criação de aplicativos web para análise de dados de forma extremamente rápida. Com ela, você pode transformar *scripts* de dados em *web apps* compartilháveis com poucas linhas de código. Ou seja, é uma ferramenta poderosa para cientistas de dados, analistas e desenvolvedores que desejam visualizar e interagir com seus dados de maneira dinâmica, sem a necessidade de conhecimento avançado em desenvolvimento web.

### 6.1.6 Python

O *Python* se tornou uma das linguagens de programação mais populares do mundo nos últimos anos. Isso se deve, principalmente, à sua versatilidade: ele funciona para o aprendizado de máquinas, construção de sites e até para automação de tarefas e testes de *softwares*.

## 6.2 ASSISTENTES

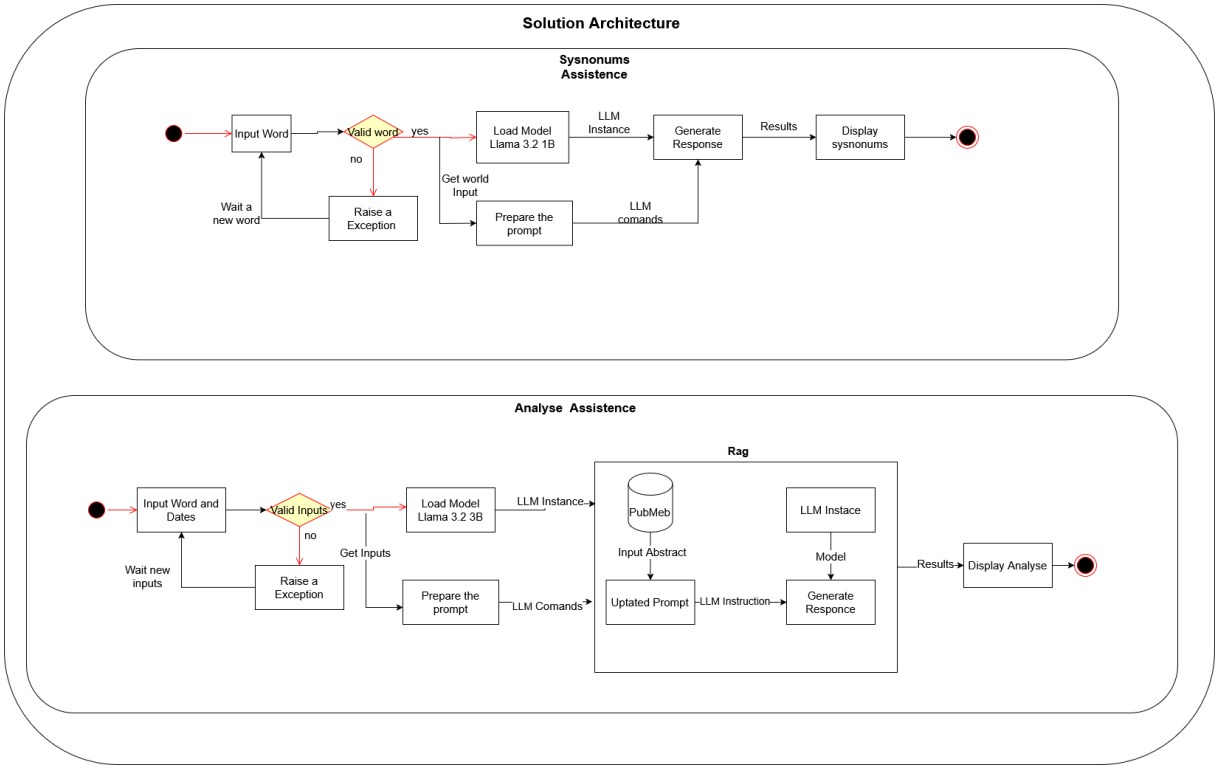
### 6.2.1 Assistente de sinônimos

O fluxo do assistente é iniciado quando o usuário passa o termo para ser consultado pelo assistente, é inicialmente verificado a validade desse termo se está vazio ou contem numeração no termo passado resultado em uma exceção ao usuário requerendo uma palavra valida, em seguida é carregado o modelo LLM *Llama 3.2 1B* e modifica o *prompt* adicionando o termo passado pelo usuário nele, o *prompt* possui os comandos que a LLM tem que seguir nesse caso gerar 10 sinônimos ao termo passado pelo usuário, no final gerado uma tabela com os resultados obtidos finalizando a execução.

### 6.2.2 Assistente de análise

O fluxo do assistente é iniciado passando a string de busca com período que essa busca será realizada e o número de artigos a ser retornados com o mínimo 2 e no máximo 200, é inicialmente verificado se as entradas são validas verificando se a string de busca está vazia, ou se a data de início e maior que a data de final da busca. Em seguida é carregado o modelo de *LLM Llama 3.2 3B* e modificado o *prompt* adicionando a string de busca passada pelo usuário, o *prompt* possui comandos contendo o passo a passo que a *LLM* tem que seguir para medir a relevância do artigo passado, em seguida a *RAG*, realiza uma pesquisa no repositório de artigos da *Pubmed* utilizando a string de busca mais o período passado para selecionar um número de artigo, os quais serão passados para *LLM* e o *prompt* para julgar a relevância deles, no final será gerado uma tabela ordenado de forma decrescente em relação à nota atribuída ao artigo com uma explicação a nota, o nome do artigo e seu link no site da *Pubmed*.

**Figura 6 –** Fluxo dos assistentes




**Fonte**: Elaborada pelo autor, 2025.

## 6.3 IMPLEMENTAÇÃO DOS ASSISTENTES

A aplicação é composta por 2 assistentes, o assistente de sinônimos e o assistente de análise. Essa sessão vai descrever todos os recursos utilizados pelos assistentes da aplicação.

### 6.3.1 Página Inicial

Componente que contém a página principal dos assistentes, contendo uma *box* de seleção para o usuário escolher o assistente que deseja utilizar, a principal biblioteca utilizada é *streamlit* descrita na sessão 6.1.

**Quadro 1 –** Página Principal

```
# −*− coding: utf−8 −*−
"""
Editor Spyder

Este      um arquivo de script  temporrio .
"""
```

```
import streamlit as st


st.set_page_config(
    page_title="Systematic Review Assistant",
    page_icon="        ",
)

st.write("# Welcome to Systematic Review Assistant!        ")

st.sidebar.success("Select a Assistant above.")

st.markdown(
    """
    Systematic Review Assistant is a tool that uses Llama as an auxiliary to perform string searches.
    **          Choose an Assistant from the sidebar**

    ### Available Assistant

    - **Synonym Assist:** Enter a term, and this tool will produce ten synonyms for it.
    - **Abstract Assistance:** Run a string search in **Pubmeb** and examine the abstract of a
        publication in English that is relevant to the Systematic Review.

"""
)
```

**Fonte**: Elaborada pelo autor, 2025.

### 6.3.2 Extração de artigos

*Script* visando extrair artigos da base de dados da *PubMed*, passando a *string* de busca do usuário para a biblioteca metapub descrita na sessão 6.1 e transformando em um *dataset* pandas.

- **Pandas:** Biblioteca de *software* criada para a linguagem *Python* para manipulação e análise de dados. Em particular, oferece estruturas e operações para manipular tabelas numéricas e séries temporais.

**Quadro 2 –** Pubmed Abstract

```
# -*- coding: utf-8 -*-
"""
Created on Wed Jun 4 05:57:32 2025
```

```
@author: Jordan
"""

import pandas as pd
from metapub import PubMedFetcher


def search_pubmed(keywords, num_of_articles = 2):
    fetch  = PubMedFetcher(api_key = '9190c5c830500523ea9c462e871226222007')
    pmids = fetch.pmids_for_query(keywords, retmax=int(num_of_articles*2))


     list_result  = []
    for  pmid in pmids:
         article  = fetch.article_by_pmid(pmid)
         list_result .append({"title":  article . title ,
                              "abstracts":  article .abstract,
                              "authors":  article .authors,
                              "keyword": article.keywords,
                              "year":  article .year,
                              "journal":  article .journal,
                              "link":  "https :// pubmed.ncbi.nlm.nih.gov/"+pmid+"/",
                              "mesh": article .mesh})

    df= pd.DataFrame(list_result)
    df  = df.dropna(subset=["abstracts"])
    return  df[:num_of_articles]
```

**Fonte**: Elaborada pelo autor, 2025.

### 6.3.3 Manipulação da LLM

*Script* visando carregar o modelo *Llama 3.2*,manipular a resposta gerada pela *LLM* e transformação da resposta para o formato *Json*.

- **Json:** Biblioteca para permitir o *python* operar arquivo no formato *JSON (JavaScript Object Notation)*.
- **Pytorch:** Biblioteca *Python* de aprendizado de máquina de código aberto usada para implementações de aprendizado profundo, como visão computacional (usando TorchVision) e processamento de linguagem natural.

**Quadro 3 –** Llama model

```
# -*- coding: utf-8 -*-
"""
Created on Tue Jun  3 22:35:37 2025

@author: Jordan
"""
from transformers import AutoTokenizer, AutoModelForCausalLM, pipeline, TextStreamer
import torch
from torch import cuda, bfloat16
import json

device = f'cuda:{cuda.current_device()}' if cuda.is_available() else 'cpu'


def load_model(path_model):
    tokenizer = AutoTokenizer.from_pretrained(path_model)

    model = AutoModelForCausalLM.from_pretrained(
        path_model,
        return_dict=True,
        low_cpu_mem_usage=True,
        torch_dtype=torch.float16,
        device_map= device,
        trust_remote_code=True,
    )

    if tokenizer.pad_token_id is None:
        tokenizer.pad_token_id = tokenizer.eos_token_id
    if model.config.pad_token_id is None:
        model.config.pad_token_id = model.config.eos_token_id
    return model,tokenizer


def generate_output_model(model, tokenizer, role, word ,max_token = 2400):

    pipe = pipeline(
        "text-generation",
        model=model,
        tokenizer=tokenizer,
        torch_dtype=torch.float16,
        device_map= device,
    )
    print(role)
    print(word)
    role = [role[0],{'role':role[1]['role'],
```

```
                    'content': role[1]['content'].format(word=word.strip())}]

    prompt = tokenizer.apply_chat_template(
        role, tokenize=False, add_generation_prompt=True
    )

    return pipe(prompt, max_new_tokens=max_token, do_sample=True)


def get_model_output_in_json(output_model):
    try:
        str1 = output_model[0]['generated_text'].split(" ``` ")[1].replace("json","")
        return json.loads(str1)
    except:
        print('Error in generate Json')
```

**Fonte**: Elaborada pelo autor, 2025.

### 6.3.4 Synonyms Assistence

Componente que contém a página de assistente de sinônimos, onde será retornada uma lista contendo 10 sinônimos para a palavra passada pelo usuário.

- **Re:** Este módulo fornece operações para correspondência de expressões regulares semelhantes às encontradas em Perl. O nome do módulo vem das iniciais do termo em inglês regular expressions, também frequentemente chamadas de regex.

**Quadro 4 –** Synonyms Assistence

```
# -*- coding: utf-8 -*-
"""
Created on Wed Jun 4 05:58:16 2025

@author: Jordan
"""

from auxiliary.variables import prompt_synonyms, path_llama_3_2_1b
import auxiliary.llama_model as llama_model
import pandas as pd
import streamlit as st
import re

st.title("Synonyms Assistence")
```

```
model_1b, token_1 = llama_model.load_model(path_llama_3_2_1b)

synonym_word = st.text_input("Type the word you wish as a synonym: ", "AI")
if  len(synonym_word) <= 0:
    st.error('Empty a word')
elif  re.search('[0-9]', synonym_word):
    st.error('Put a  valid  word')
else:
    result_synonym = llama_model.generate_output_model(model_1b, token_1, prompt_synonyms,
        synonym_word)
    df_synonym = pd.DataFrame(llama_model.get_model_output_in_json(result_synonym))
    df_synonym.columns = ['Synonym']
    st.table(df_synonym)
```

**Fonte**: Elaborada pelo autor, 2025.

Composto que contém a função de assistente de análise, onde o usuário insere sua *string* de busca e o assistente retorna uma lista com esses artigos ordenados pela nota atribuída a ele, possui um limite de retornar no máximo 200 artigos.

### 6.3.5 Analyze Assistence

**Quadro 5 –** Analyze Assistence

```
# -*- coding: utf-8 -*-
"""
Created on Wed Jun 4 06:40:17  2025

@author: Jordan
"""

import streamlit  as  st
from auxiliary.variables  import keywords,path_llama_3_2_3b,prompt_analyze
import auxiliary.llama_model as llama_model
from auxiliary.pubmed_abstract import search_pubmed
import re
import pandas as pd


st. title ("Analyze Assistence")



string_search  = st.text_input("Type your String Search:  ",  "")
```

```
select_begin_date = st.date_input('Begin Period of Search: ')
select_end_date = st.date_input('End Period of Search: ')

num_articles = st.slider("Number of articles", 0, 200, 2)

if len(string_search) <= 0:
    st.error('Empty a word')
elif re.search('[0-9]', string_search):
    st.error('Put a valid word')
elif select_begin_date >= select_end_date:
    st.error('Invalid range of date')
else:


        key = keywords.format(begin_date = select_begin_date,
                              end_date = select_end_date,
                              string_search = string_search)
        df = search_pubmed(key,num_articles )
        print(df.shape)

        if df.shape[0]<=0:
            st.error('Articles not found')

        role = [{'role':prompt_analyze[0]["role"],
                 'content':prompt_analyze[0]["content"].replace('string_search',string_search)},prompt_analyze[1
        model_3b, token_3 = llama_model.load_model(path_llama_3_2_3b)

        if model_3b == None or token_3 ==None:
            st.error('Error load a model')

        list_result = []
        for abstract in df[['title','link','abstracts']].values:
            dict_r ={}
            result_synonym = llama_model.generate_output_model(model_3b, token_3, role,
                abstract[2])
            r = llama_model.get_model_output_in_json(result_synonym)
            dict_r['Title'] = abstract [0]
            dict_r['link'] = abstract[1]
            dict_r['Explain'] = r[ list (r.keys()) [1]]
            dict_r['score'] = r[ list (r.keys()) [0]]
            list_result .append(dict_r)
        df_r = pd.DataFrame(list_result)
        if df_r.shape[0]<=0:
            st.error('Empty results')
        df_r = df_r[['Title','score','link','Explain']]
        df_r = df_r.sort_values('score', ascending= False)
```

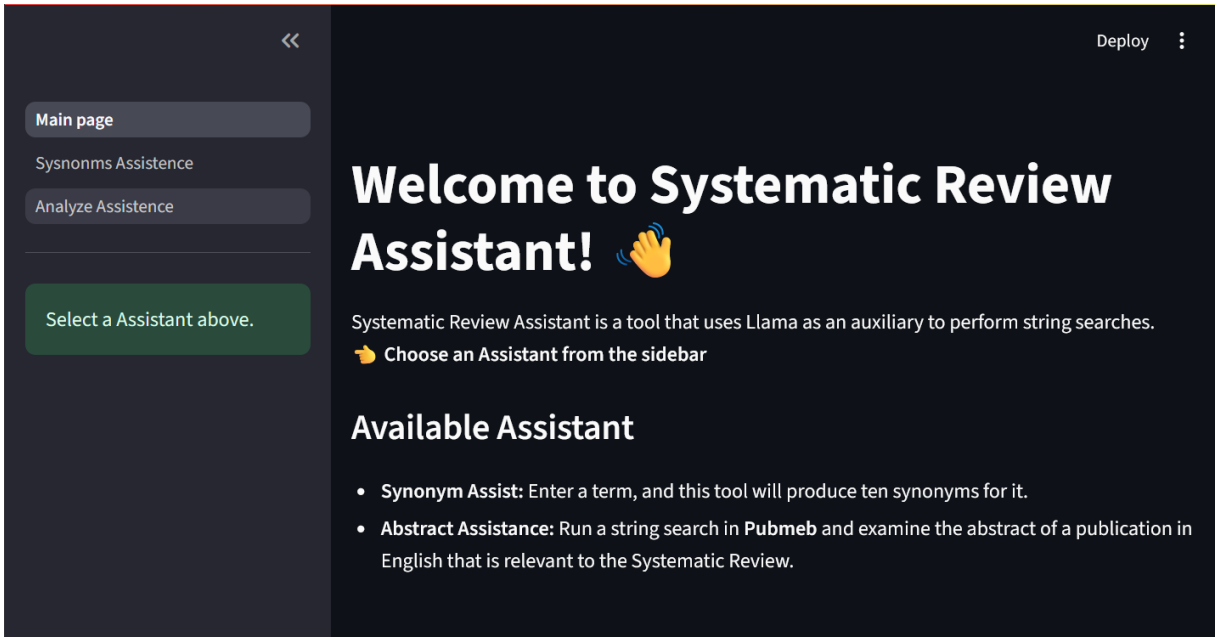
```
        st.dataframe(df_r, use_container_width=True)
```

**Fonte**: Elaborada pelo autor, 2025.

## 6.4 APLICAÇÃO

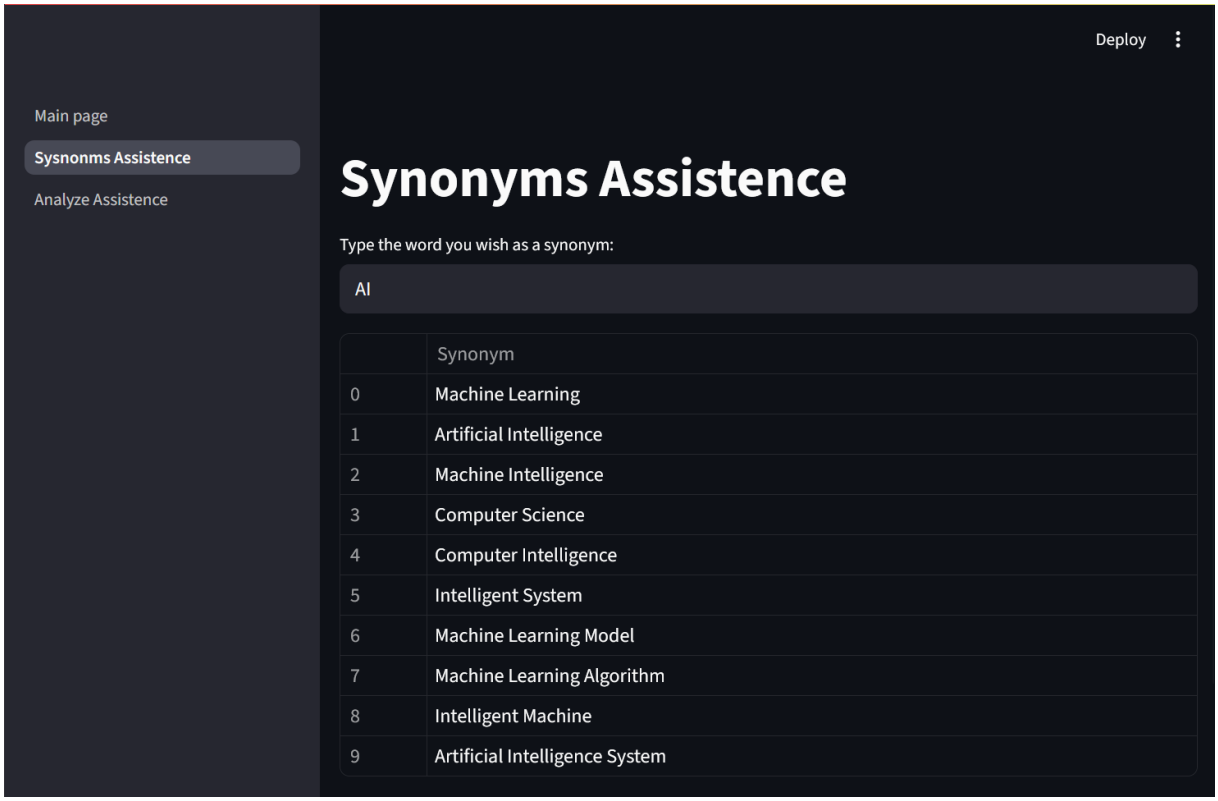
A interface da solução assistente de revisão sistemática é feita com a biblioteca *Streamlit*, ferramenta poderosa que permite a criação de web apps de maneira rápida e intuitiva. Ele é especialmente útil para cientistas de dados, analistas e qualquer pessoa que queira transformar scripts *Python* em aplicativos interativos sem precisar de conhecimentos avançados em desenvolvimento web, contendo uma página principal explicada aplicação e uma aba lateral contendo os assistentes. Os assistentes utilizam modelo *Llama* com a versão 3.2 pertencente a empresa Meta, mas tem licença gratuita, para carregar os modelos é utilizado a biblioteca transformers, a base de dados dos artigos é alimentada utilizado a biblioteca metapub permite baixa os elementos de artigos como abstract, título, *link* do artigo, palavras-chave dentre outros presentes no banco de dados de artigos da *Pubmed*, e os modelos com prompts e a base de dados gerada são utilizados pela *RAG* para aumentar o conhecimento presente na *LLM* em relação ao assunto pesquisado pelo autor e por fim são gerados os resultados de cada assistente.

**Figura 7 –** Página inicial da solução




**Fonte**: Elaborada pelo autor, 2025.

**Figura 8 –** Página da assistência de sinônimos




**Fonte**: Elaborada pelo autor, 2025.

**Figura 9 –** Página da assistência de analise


**Fonte**: Elaborada pelo autor, 2025.

# 7 CONSIDERAÇÕES

A presente aplicação foi desenvolvida com o objetivo de auxiliar pesquisadores na etapa inicial de uma revisão sistemática, em especial na construção da string de busca e na seleção preliminar de artigos relevantes. Durante o processo de desenvolvimento e validação da ferramenta, foi possível identificar tanto seus pontos fortes quanto suas limitações, o que aponta para diversas possibilidades de aprimoramento futuro.

Uma limitação significativa da aplicação atual é sua capacidade de realizar buscas apenas em artigos escritos em língua inglesa. Embora o inglês seja a língua predominante na literatura científica, a inclusão de artigos em outras línguas poderia ampliar significativamente o escopo da revisão sistemática e permitir análises mais inclusivas e representativas. Contudo, essa ampliação requer uma avaliação cuidadosa da capacidade dos modelos de linguagem de grande escala (LLMs) em compreender outras línguas, uma vez que esses modelos geralmente são treinados com um volume muito maior de textos em inglês.

Outro ponto a ser considerado é a tecnologia utilizada na construção da interface da aplicação. Embora o uso do *Streamlit* tenha se mostrado eficiente para o desenvolvimento de dashboards de forma rápida e intuitiva, essa biblioteca apresenta limitações quando se busca uma solução mais robusta e escalável. Caso se deseje uma aplicação com maior nível de personalização, escalabilidade e integração com outros sistemas, seria recomendada a migração para outras bibliotecas ou frameworks de desenvolvimento web mais completos, como *Flask*, *Django* ou mesmo ferramentas em *JavaScript*, como *React* ou *Vue.js*.

Além disso, a aplicação atualmente se limita ao uso da base de dados PubMed para a busca e recuperação dos artigos. Essa escolha se deu pelas restrições de acesso a outras bases amplamente utilizadas em revisões sistemáticas, como a Scopus, que requerem conta institucional ou assinaturas pagas. Esse fator constitui uma limitação importante, pois restringe a abrangência dos resultados obtidos. Como possível solução, pode-se considerar a integração com outras fontes de dados gratuitas ou a implementação de mecanismos que permitam aos usuários, quando possível, autenticar-se com contas institucionais para acessar bases pagas.

Em resumo, a ferramenta desenvolvida oferece uma contribuição relevante ao automatizar e facilitar parte do processo de revisão sistemática. Entretanto, para alcançar maior abrangência, flexibilidade e eficiência, são necessárias melhorias relacionadas à ampliação linguística, à modernização da interface e à diversificação das fontes de dados utilizadas.

# 8 CONCLUSÕES E TRABALHOS FUTUROS

Neste trabalho foi apresentada dois assistentes voltados para a revisão sistemática: um para a geração automatizada de sinônimos, que contribui para a construção e refinamento das *strings* de busca, e outro para a análise de *abstracts*, oferecendo ao pesquisador uma classificação da relevância dos artigos com base em notas e justificativas. Os assistentes apresentados tem como objetivo agilizar e auxiliar os autores de revisão sistemática com foco na elaboração da string de busca e seleção de artigos relevante a pesquisa feita pelo autor.

As funcionalidades dos assistentes foram detalhadas, bem como utilizados de acordo com a necessidade do usuário. Espera-se que, apesar da falta de testes intensivos dos assistentes e de uma literatura que trate diretamente da aplicação do método *RAG* em revisões sistemáticas, os resultados desta dissertação indicam que essa é uma direção viável e com grande potencial. A flexibilidade do método *RAG*, já observada em diversas outras áreas, aliada ao poder das *LLMs*, mostra-se eficaz também neste novo contexto isso demostrado na revisão de escopo feito anteriormente.

Embora não tenha sido possível validar a ferramenta, alguns experimentos iniciais dentro do próprio grupo de pesquisa demonstaram que a ferramenta mostra-se eficaz nas etapas iniciais da revisão sistemática. Relatos informais dos pesquisadores, que estão executando revisões sistemáticas, indicam que se durante a definição do protocolo, os mesmos tivessem acesso à ferramenta, novos termos teriam sido incluídos na busca. Os pesquisadores relataram ainda, que a identificação e marcação de artigos relevantes retornados pela busca, facilita muito a etapa inicial de *screening* e definição de artigos a serem usados em grupo de controle.

Conforme pode ser observado, a aplicação desenvolvida nesta pesquisa apresenta um potencial considerável de expansão para outras etapas da revisão sistemática, em especial às etapas de *screening* e extração de informações dos artigos selecionados. Entretanto, para isso faz-se necessário o acesso aos artigos completos e acesso a mais recusros computacionais para processamento dos dados.

Portanto, os próximos passos desta pesquisa envolvem o desenvolvimento de técnicas complementares de segmentação textual, identificação de seções relevantes e adaptação estrutural dos documentos, com vistas à ampliação do uso de *LLMs* para uma atuação mais abrangente em todas as fases da revisão sistemática. Isso inclui possível inegração com ferramentas voltadas a execução de reviões sistemáticas.

Mediante ao que foi exposto, destaca-se que apesar das limitações atuais, a ferramenta mostra-se útil para o proposito ao qual foi criada e que esse é apenas o passo

inicial para o desenvolvimento de um sistema mais abrangente e completo, para facilitar o processo de revisões e avaliação de tecnologias em saúde.

# REFERÊNCIAS BIBLIOGRÁFICAS

atrick *et al.* 2021ATRICK; LEWIS, E.; PEREZ, A.; PIKTUS, F.; PETRONI, V.; KARPUKHIN, N.; GOYAL, H.; KüTTLER, M.; LEWIS, W.-t.; YIH, T.; ROCKTäSCHEL, S.; RIEDEL, D.; KIELA. Retrieval-augmented generation for knowledge-intensive nlp tasks. **NIPS: Neural Information Processing Systems**, 2021.

Cheonsu e Jeong 2023CHEONSU; JEONG. Generative ai services using an enterprise data-based application architecture. **Advances in Artificial Intelligence and Machine Learning**, 2023.

David *et al.* 2020DAVID; MOHER, A.; LIBERATI, J.; TETZLAFF, D.; G.; ALTMAN. **PRISMA checklist 2020**. 2020. Disponível em: <https://static1.squarespace.com/static/65b880e13b6ca75573dfe217/t/67ad313f1c80aa5235fce0d0/1739403584136/PRISMA_2020_checklist.pdf>. Acesso em: 20 fev. 2025.

Gihan *et al.* 2024GIHAN; GAMAGE, N.; MILLS, D.; DE; SILVA, M.; MANIC, H.; MORALIYAGE, A.; JENNINGS. Multi-agent rag chatbot architecture for decision support in net-zero emission energy systems. **IEEE International Conference on Industrial Technology (ICIT)**, 2024.

Iain *et al.* 2002IAIN; CHALMERS, L.; V; HEDGES, H.; COOPER. A brief history of research synthesis. **Evaluation  the Health Professions**, 2002.

Jean *et al.* 2023JEAN; CARLOS; BORGES; BRITO, D.; LOPES; MARTINS. Revisão sistemática da literatura na ciência da informação: uma descrição detalhada dos passos metodológicos. **InCID: Revista de Ciência da Informação e Documentação**, 2023.

Jin *et al.* 2024JIN; GE, S.; SUN, J.; OWENS, V.; GALVEZ, O.; GOLOGORSKAYA, J.; C; LAI, M.; J; PLETCHER, K.; LAI. Development of a liver disease-specific large language model chat interface using retrieval-augmented generation. **Journal of Hepatology**, 2024.

Keith, D. e Foote 2024KEITH; D.; FOOTE. **A Brief History of Generative AI**. [S.l.]: DataVersity, 2024.

Marcos *et al.* 2009MARCOS; R; DE; SOUSA, A.; LUIZ; P; RIBEIRO. Systematic review and meta-analysis of diagnostic and prognostic studies: a tutorial. **ABC Cardiol**, 2009.

Marcus e Almeida 2023MARCUS; ALMEIDA. **Machine Learning: o que é aprendizado semi-supervisionado**. 2023. Disponível em: <https://www.alura.com.br/artigos/machine-learning-aprendizado-semi-supervisionado>. Acesso em: 10 jan. 2025.

Martin *et al.* 2017MARTIN; J; WESTGATE, D.; B; LINDENMAYE. The difficulties of systematic reviews. **Society for Conservation Biology**, 2017.

Sonia *et al.* 2024SONIA; VAKAYIL, S.; JULIET ANITHA, S.; VAKAYIL. Rag-based llm chatbot using llama-2. **International Conference on Devices, Circuits and Systems (ICDCS)**, 2024.

Stewart *et al.* 2024STEWART; KIRUBAKARAN, J.; WILSIE; KATHRINE, G.; MARY; KANAGA, M.; RAJA, R.; GINO; SINGH, Y. A rag-based medical assistant especially for infectious diseases. **International Conference on Inventive Computation Technologies (ICICT)**, 2024.

Taís *et al.* 2014TAíS; FREIRE; GALVãO, M.; GOMES; PEREIRA. Revisões sistemáticas da literatura: passos para sua elaboração. **Epidemiologia e Serviços de Saúde**, 2014.

Tim *et al.* 2023TIM; DETTMERS, A.; PAGNONI, A.; HOLTZMAN, L.; ZETTLEMOYER. Qlora: efficient finetuning of quantized llms. **NIPS - Neural Information Processing Systems Foundation**, 2023.

Yoshua *et al.* 2003YOSHUA; BENGIO, R.; DUCHARME, P.; VINCENT, C.; JAUVIN. A neural probabilistic language model. **Journal of Machine Learning Research**, 2003.

# APÊNDICE A - PRISMA 2020 CHECKLIST

**Tabela 3 –** PRISMA 2020 Checklist- Parte 1

| Section and Topic | Item | Checklist Item | Location where item is reported |
|---|---|---|---|
| TITLE | | | |
| Title | 1 | Identify the report as a systematic review. | |
| ABSTRACT | | | |
| Abstract | 2 | See the PRISMA 2020 for Abstracts Checklist. | |
| INTRODUCTION | | | |
| Rationale | 3 | Describe the rationale for the review in the context of existing knowledge. | |
| Objectives | 4 | Provide an explicit statement of the objective(s) or question(s) the review addresses. | |
| METHODS | | | |
| Eligibility Criteria | 5 | Specify the inclusion and exclusion criteria for the review and how studies were grouped for the syntheses | |
| Information sources | 6 | Specify all databases, registers, websites, organisations, reference lists and other sources searched or consulted to identify studies. Specify the date when each source was last searched or consulted. | |
| Search strategy | 7 | Present the full search strategies for all databases, registers and websites, including any filters and limits used. | |
| Selection process | 8 | Specify the methods used to decide whether a study met the inclusion criteria of the review, including how many reviewers screened each record and each report retrieved, whether they worked independently, and if applicable, details of automation tools used in the process. | |
| Data collection process | 9 | Specify the methods used to collect data from reports, including how many reviewers collected data from each report, whether they worked independently, any processes for obtaining or confirming data from study investigators, and if applicable, details of automation tools used in the process. | |
| Data items | 10a | List and define all outcomes for which data were sought. Specify whether all results that were compatible with each outcome domain in each study were sought (e.g. for all measures, time points, analyses), and if not, the methods used to decide which results to collect. | |
| | 10b | List and define all other variables for which data were sought (e.g. participant and intervention characteristics, funding sources). Describe any assumptions made about any missing or unclear information. | |

**Fonte**: David *et al.* (2020).

**Tabela 4 –** PRISMA 2020 Checklist - Parte 2

| Section and Topic | Item | Checklist Item | Location where item is reported |
|---|---|---|---|
| Study risk of bias assessment | 11 | Specify the methods used to assess risk of bias in the included studies, including details of the tool(s) used, how many reviewers assessed each study and whether they worked independently, and if applicable, details of automation tools used in the process. | |
| Effect measures | 12 | Specify for each outcome the effect measure(s) (e.g. risk ratio, mean difference) used in the synthesis or presentation of results | |
| Synthesis methods | 13a | Describe the processes used to decide which studies were eligible for each synthesis (e.g. tabulating the study intervention characteristics and comparing against the planned groups for each synthesis (item #5)). | |
| | 13b | Describe any methods required to prepare the data for presentation or synthesis, such as handling of missing summary statistics, or data conversions. | |
| | 13c | Describe any methods used to tabulate or visually display results of individual studies and syntheses. | |
| | 13d | Describe any methods used to synthesize results and provide a rationale for the choice(s). If meta-analysis was performed, describe the model(s), method(s) to identify the presence and extent of statistical heterogeneity, and software package(s) used. | |
| | 13e | Describe any methods used to explore possible causes of heterogeneity among study results (e.g. subgroup analysis, meta-regression). | |
| | 13f | Describe any sensitivity analyses conducted to assess robustness of the synthesized results. | |
| Reporting bias assessment | 14 | Describe any methods used to assess risk of bias due to missing results in a synthesis (arising from reporting biases). | |
| Certainty assessment | 15 | Describe any methods used to assess certainty (or confidence) in the body of evidence for an outcome. | |
| RESULTS | | | |
| Study selection | 16a | Describe the results of the search and selection process, from the number of records identified in the search to the number of studies included in the review, ideally using a flow diagram. | |
| | 16b | Cite studies that might appear to meet the inclusion criteria, but which were excluded, and explain why they were excluded. | |
| Study Characteristics | 17 | Cite each included study and present its characteristics. | |
| Risk of bias in studies | 18 | Present assessments of risk of bias for each included study. | |
| Results of individual studies | 19 | For all outcomes, present, for each study: (a) summary statistics for each group (where appropriate) and (b) an effect estimate and its precision (e.g. confidence/credible interval), ideally using structured tables or plots. | |

**Fonte**: David *et al.* (2020).

**Tabela 5 –** PRISMA 2020 Checklist- Parte 3

| Section and Topic | Item | Checklist Item | Location where item is reported |
|---|---|---|---|
| Results of syntheses | 20a | For each synthesis, briefly summarise the characteristics and risk of bias among contributing studies | |
| | 20b | Present results of all statistical syntheses conducted. If meta-analysis was done, present for each the summary estimate and its precision (e.g. confidence/credible interval) and measures of statistical heterogeneity. If comparing groups, describe the direction of the effect. | |
| | 20c | Present results of all investigations of possible causes of heterogeneity among study results. | |
| | 20d | Present results of all sensitivity analyses conducted to assess the robustness of the synthesized results. | |
| Reporting biases | 21 | Present assessments of risk of bias due to missing results (arising from reporting biases) for each synthesis assessed. | |
| Certainty of evidence | 22 | Present assessments of certainty (or confidence) in the body of evidence for each outcome assessed. | |
| DISCUSSION | | | |
| Discussion | 23a | Provide a general interpretation of the results in the context of other evidence. | |
| | 23b | Discuss any limitations of the evidence included in the review. | |
| | 23c | Discuss any limitations of the review processes used. | |
| | 23d | Discuss implications of the results for practice, policy, and future research. | |
| OTHER INFORMATION | | | |
| Registration and protocol | 24a | Provide registration information for the review, including register name and registration number, or state that the review was not registered. | |
| | 24b | Indicate where the review protocol can be accessed, or state that a protocol was not prepared. | |
| | 24c | Describe and explain any amendments to information provided at registration or in the protocol. | |
| Support | 25 | Describe sources of financial or non-financial support for the review, and the role of the funders or sponsors in the review. | |
| Competing interests | 26 | Declare any competing interests of review authors. | |
| Availability of data, code and other materials | 27 | Report which of the following are publicly available and where they can be found: template data collection forms; data extracted from included studies; data used for all analyses; analytic code; any other materials used in the review. | |

**Fonte**: David *et al.* (2020).