Multi-Agent Systems and Prompt Engineering in automating and optimizing prompt generation and refinement processes: a Systematic Literature Review

Allan M. Gonçalves[1], **Cleo Z. Billa**[1], **Diana F. Adamatti**[1]

[1]Centro de Ciências Computacionais – Universidade Federal do Rio Grande (FURG)
Caixa Postal 474 – 96201-900 – Rio Grande – RS – Brazil

{Allanmgoncalves00, cleo.billa, dianaada}@gmail.com

Abstract. *This Systematic Literature Review investigated how Multi-Agent Systems have been employed to automate and optimize prompt generation and refinement processes. The review was conducted according to the PRISMA2020 guidelines and a research protocol based on the PICO framework. A total of 39 studies were initially identified, from which 6 primary studies were selected after the screening and eligibility phases. The review concludes that the integration of specialized agents, iterative refinement strategies, and structured evaluation mechanisms constitutes a promising paradigm for advancing Prompt Engineering and improving the reliability, effectiveness, and alignment of LLM-generated outputs.*

Resumo. *Esta Revisão Sistemática da Literatura investigou como Sistemas Multiagentes têm sido aplicados na Engenharia de Prompts para otimizar e automatizar a interação com Grandes Modelos de Linguagem (LLMs). O estudo foi conduzido seguindo as diretrizes PRISMA2020 e um protocolo baseado na estratégia PICO, com buscas em bases científicas consolidadas consolidadas. Inicialmente, foram encontrados 39 estudos, dos quais apenas 6 atenderam plenamente aos critérios de inclusão. Conclui-se que a integração entre sistemas multiagentes, refinamento iterativo e mecanismos estruturados de avaliação representam uma evolução para a Engenharia de prompts, contribuindo para a obtenção de respostas mais precisas e alinhadas às intenções dos usuários.*

1. Introdução

Nos últimos anos, observa-se um crescimento exponencial no uso de sistemas baseados em Inteligência Artificial Generativa (IA), impulsionado principalmente pela popularização de grandes modelos de linguagem (Large Language Models - LLMs), como aqueles utilizados em aplicações conversacionais e ferramentas de geração de conteúdo [Vaswani et al. 2023]. Esse fenômeno, descrito como um "boom" da IA generativa, está relacionado à capacidade desses modelos de produzir conteúdos, incluindo textos, imagens, áudios e vídeos, a partir de dados de entrada fornecidos pelos usuários. Tal avanço tem impactado significativamente diferentes áreas, desde educação e desenvolvimento de *software* até *marketing* e produção criativa, consolidando a IA generativa como uma das principais tendências tecnológicas contemporâneas [Zhao et al. 2026].

Entretanto, apesar do enorme potencial dessas tecnologias, a qualidade das respostas geradas por esses modelos de linguagem depende fortemente da forma como as

instruções são fornecidas. Nesse contexto, surge a engenharia de *prompts* (*prompt engineering*), prática de projetar e estruturar entradas de forma estratégica para orientar os modelos a produzirem resultados mais precisos e alinhados ao objetivo do usuário [Sahoo et al. 2025]. Estudos recentes destacam que pequenas variações na formulação de um prompt [Jason Wei and Zhou 2022], como a definição de contexto, papel do agente ou exemplos fornecidos, podem impactar significativamente a qualidade, a coerência e a utilidade das respostas geradas.

Apesar dos avanços na área, a elaboração de *prompts* eficazes ainda representa um desafio, especialmente para usuários que não possuem conhecimento ou experiência, que frequentemente fornecem instruções vagas, ambíguas ou incompletas. Isso evidencia uma lacuna entre o potencial dos modelos de IA generativa e a capacidade dos usuários em explorá-los de forma eficiente. Nesse contexto, [Guo et al. 2024] destacam que sistemas multiagentes baseados em LLMs têm se mostrado uma abordagem promissora para a resolução de tarefas complexas, uma vez que permitem distribuir responsabilidades entre agentes especializados, capazes de colaborar em atividades como planejamento, avaliação, refinamento e tomada de decisão. Dessa forma, torna-se possível o desenvolvimento de abordagens automatizadas com o uso de agentes que auxiliem na construção de *prompts* mais estruturados, possibilitando a obtenção de respostas mais precisas e confiáveis.

Diante deste cenário, este estudo buscou realizar uma Revisão Sistemática da Literatura (RSL) sobre abordagens automatizadas utilizando técnicas de Engenharia de Prompt com ênfase em obter-se respostas com maior precisão de LLMs. Foram conduzidas buscas em bases consolidadas, para selecionar as pesquisas mais recentes e relevantes sobre o tema de interesse deste estudo.

Este artigo está estruturado em 4 seções. Na seção 2 é apresentada a elaboração da RSL. A sumarização e síntese dos estudos está presente na seção 3. E por fim, a seção 4 apresenta a conclusão da pesquisa.

2. Metodologia

A RSL foi conduzida com o objetivo de identificar, reunir, sintetizar e analisar estudos relacionados ao uso de agentes para geração, refinamento e otimização automatizada de *prompts*, com ênfase em trabalhos que apresentem evidências empíricas. Com base nas diretrizes propostas por [Nakagawa et al. 2017], foi definido o protocolo de pesquisa, contemplando a formulação da questão de pesquisa, identificação dos estudos, critérios de seleção e extração de dados. Para garantir a transparência e rastreabilidade do processo de identificação, triagem, elegibilidade e inclusão dos estudos, foi adotado o protocolo PRISMA2020 (*Preferred Reporting Items for Systematic Reviews and Meta-Analyses*) [Page et al. 2021].

2.1. Questão de Pesquisa

Segundo [Kitchenham and Charters 2007], a elaboração da questão de pesquisa é a etapa mais importante de uma Revisão Sistemática, visto que ela guiará toda a condução do estudo. Para isso, é necessário extrair a essência da linha de pesquisa e redigi-la em formato interrogativo [Nakagawa et al. 2017]. Diante deste contexto, o presente trabalho buscou

responder à seguinte questão principal: **"Como os Sistemas Multiagentes têm sido aplicados na Engenharia de *prompts* para a otimização e o refinamento automatizado de instruções em Grandes Modelos de Linguagem?"**.

Para a estruturação da questão de pesquisa, adotou-se a estratégia PICO. Originalmente surgiu na área da saúde [Pai et al. 2004], porém foi adaptada para o contexto da Engenharia de Software e toda a área da computação como um todo [Kitchenham and Charters 2007]. O acrônimo orienta a definição dos elementos centrais da investigação, sendo divididos em:

- **P (População)**: Representa um papel específico da Engenharia de Software ou uma área de aplicação.
- **I (Intervenção)**: Refere-se ao método, ferramenta, procedimento ou tecnologia a ser investigada.
- **C (Comparação)**: Indica o método, ferramenta, procedimento ou tecnologia com o qual a intervenção deve ser comparada.
- **O (Outcomes/Resultados)**: Correspondem aos fatores considerados importantes para caracterizar o que está sendo investigado.

Com base nestas definições, a Tabela 1 apresenta a aplicação dos critérios PICO para a formulação da questão Primária deste estudo.

Tabela 1. Elementos definidos pela estratégia PICO para a RSL.

Critério	Descrição
População	Grandes Modelos de Linguagem (LLMs)
Intervenção	Sistemas Multiagentes
Comparação	Engenharia de Prompt (manual)
Resultados	Otimização e refinamento automatizado de instruções

2.2. Identificação de Estudos

Após a definição da questão de pesquisa, iniciou-se a etapa de identificação dos estudos, correspondente à primeira fase do protocolo PRISMA. Essa etapa tem como objetivo localizar e selecionar trabalhos relevantes ao tema investigado por meio de buscas em bases de dados científicas. Inicialmente, foram definidas as palavras-chave a partir dos principais conceitos presentes na questão de pesquisa, esses termos representam os elementos centrais do estudo. As palavras-chave selecionadas foram: `Sistemas Multiagentes`, `Engenharia de Prompt`, `Grandes Modelos de Linguagem`, `Modelo Baseado em Agentes`, `Refinamento de Prompt`.

Com base nessas palavras-chave e em termos correlatos identificados na literatura, foi elaborada a *string* de busca. Para sua elaboração, utilizaram-se operadores lógicos e caracteres especiais que permitem combinar os conceitos de interesse e restringir os resultados aos estudos alinhados ao objetivo da pesquisa. Os recursos utilizados foram:

- **Parênteses ()**: Servem para agrupar expressões booleanas compostas e definir a ordem de avaliação dos termos.
- **Aspas ("")**: Utilizadas para tratar uma expressão composta por múltiplas palavras como um único termo de busca.

- **Asterisco (*)**: Funciona como um caractere de truncamento, retornando todas as variações do sufixo de uma palavra (por exemplo, iterat* pode retornar *iteration* ou *iterate*).
- **AND**: Operador de interseção. Combina diferentes expressões e exige que todos os termos separados por ele estejam presentes no registro retornado.
- **OR**: Operador de união. Utilizado para incluir sinônimos ou termos relacionados, retornando registros que contenham qualquer uma das expressões associadas.

A estratégia de busca foi documentada de forma a garantir a transparência, rastreabilidade e reprodutibilidade do processo, conforme recomendado pelas diretrizes do protocolo [Nakagawa et al. 2017]. A Figura 1 evidencia o resultado da elaboração da *string* de busca.

(("LLM agent" OR "autonomous agent" OR "based-model agent" OR "LLM-based agent") AND ("automatic prompt engineering" OR "automatic prompt optimization" OR "prompt optimization" OR "prompt refinement") AND (iterat OR self-refin* OR feedback OR "human-in-the-loop") AND (evaluat* OR accuracy OR "response quality"))*

Figura 1. String de busca utilizada na RSL.

A partir da *string* de busca, selecionaram-se quatro bases de dados para a condução das pesquisas e identificação dos estudos investigados. O critério central para a essa escolha baseou-se na confiabilidade, abrangência e relevância dessas bases na comunidade científica. A Tabela 2 apresenta as bases de dados utilizadas neste RSL, juntamente com suas áreas de cobertura e as justificativas para sua seleção. A utilização de diferentes fontes teve como objetivo ampliar a abrangência das buscas, permitindo maior cobertura sobre estudos relacionados com a questão da pesquisa.

Tabela 2. Bases de dados utilizadas na RSL.

Base de dados	Área de cobertura	Justificativa
Scopus	Multidisciplinar	Uma das maiores bases de resumos e citações da literatura com revisão por pares, garante uma ampla cobertura de periódicos e anais de conferências científicas de alto impacto.
IEEE Xplore	Computação, Engenharia e Tecnologia	Relevante para pesquisas envolvendo Inteligência Artificial, Sistemas Multiagentes e Engenharia de Software.
SpringerLink	Multidisciplinar	Disponibiliza periódicos e conferências com publicações recentes sobre IA e Ciência da Computação.
arXiv	Multidisciplinar	Permite acesso a pesquisas recentes e trabalhos ainda não publicados formalmente.

2.3. Critérios de Inclusão e Exclusão

Com o objetivo de garantir a qualidade e relevância dos estudos selecionados, foram definidos critérios de inclusão e exclusão, apresentados na Tabela 3. Esses critérios orientam o processo de seleção dos estudos, permitindo a eliminação de publicações incompatíveis em etapas iniciais da revisão. Dessa forma, busca-se reduzir o conjunto de resultados recuperados, assegurando que apenas estudos que apresentam maior alinhamento com a questão de pesquisa avancem para a etapa de avaliação completa e extração de dados.

Tabela 3. Critérios de inclusão e exclusão adotados na RSL.

	Critério de inclusão
CI1	Estudos que abordem agentes baseados em LLMs ou sistemas multiagentes aplicados à geração, otimização ou refinamento iterativo e automatizado de prompts.
CI2	Estudos que apresentem métodos, métricas ou frameworks de avaliação de qualidade, precisão, consistência ou confiabilidade das respostas e dos prompts gerados.
CI3	Estudos que descrevem mecanismos, arquiteturas ou fluxos de trabalho relacionados à interação entre agentes, modelos de linguagem e processos de refinamento de prompts.
CI4	Estudos publicados a partir de 2024.
CI5	Estudos publicados em Inglês ou Português.
	Critério de exclusão
CE1	Estudos fora do escopo de agentes, engenharia de prompt ou que abordem IA Generativa focada exclusivamente em domínios não textuais.
CE2	Estudos puramente teóricos que não apresentem avaliação experimental, métricas de desempenho ou análise da qualidade dos resultados.
CE3	Artigos de revisão de literatura, resumos expandidos de poucas páginas, capítulos de livros teóricos ou trabalhos sem descrição metodológica suficiente.

2.4. Processo de Seleção de Resultados

Após a execução das buscas nas bases de dados selecionadas, obteve-se um total de **39 artigos**. A Figura 2 apresenta o fluxograma de seleção elaborado com base no protocolo PRISMA2020 [Page et al. 2021], evidenciando as etapas de identificação, triagem, elegibilidade e inclusão dos estudos ao longo da revisão.

A seleção dos estudos foi conduzida de forma sistemática, utilizando os critérios de inclusão e exclusão definidos na **Seção 2.3**. Para isso, foi estabelecida uma estratégia composta por quatro etapas sequenciais: seleção inicial, leitura do título e resumo (*abstract*), leitura diagonal e leitura completa. Essa abordagem permitiu reduzir gradualmente o conjunto de resultados, mantendo apenas os trabalhos mais alinhados à questão de pesquisa.

A seleção inicial consiste na filtragem preliminar dos registros recuperados. Nessa fase, realizou-se a verificação de registros em duplicidade (indexados em múltiplas bases)

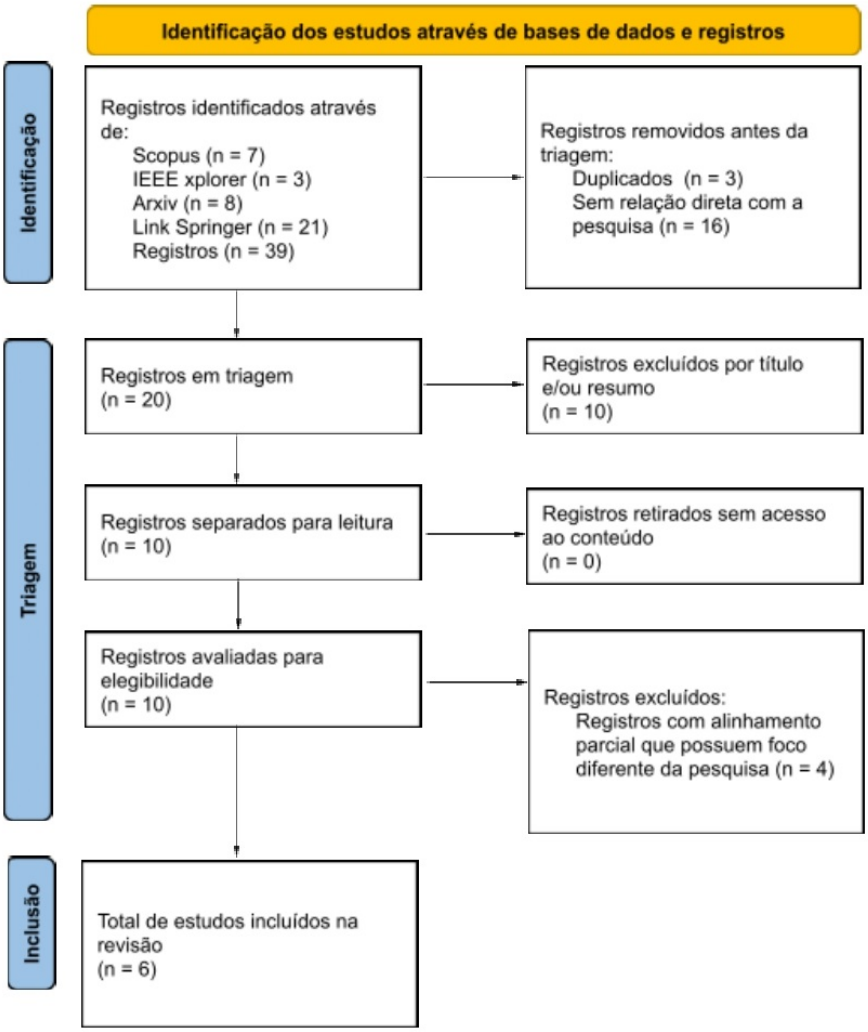

Figura 2. Etapas da Metodologia PRISMA.

e análise das palavras-chave. O objetivo foi selecionar apenas os estudos que abrangem todas áreas correlacionadas com a pesquisa. Para ilustrar esse processo, foi elaborado um diagrama de Venn (Figura 3), o qual evidencia que os registros de interesse encontram-se exatamente na interseção das áreas principais. Dos **39 estudos** inicialmente recuperados, foram identificados e removidos **3 duplicatas** e **16 artigos** por não apresentarem relação direta com o tema investigado, resultando em **20 publicações** selecionadas para a continuidade da revisão.

Na segunda etapa, foi realizada a leitura dos títulos e resumos dos estudos selecionados. Essa análise permitiu identificar rapidamente os objetivos, métodos e contribuições de cada pesquisa. Aplicando-se os critérios de inclusão, foram selecionados os trabalhos cujo foco principal estava relacionado à geração, refinamento ou otimização de *prompts*, independentemente da abordagem utilizada. Ao final dessa etapa, **10 estudos** foram considerados relevantes para uma análise mais aprofundada.

Antes da continuidade do processo, foi necessário verificar a disponibilidade do texto completo dos estudos selecionados. Confirmou-se que todos os 10 registros pos-

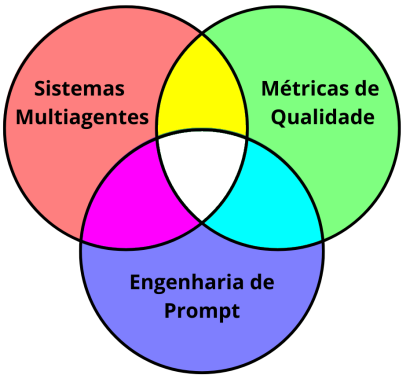

Figura 3. Diagrama de Venn com as áreas de pesquisa da RSL.

suem acesso livre. Em seguida, foi aplicada a técnica de leitura diagonal. Essa técnica consiste na leitura seletiva de seções, incluindo o título, resumo, introdução, figuras, tabelas e conclusão. Constatou-se que **4 artigos** possuíam alinhamento parcial com a pesquisa, com um foco restrito a nichos não generalizáveis. Dessa forma, esses estudos foram excluídos da análise.

Por fim, o conjunto final de estudos foi composto de **6 artigos** que possuem alta relevância com o tema investigado, estes, foram submetidos à leitura integral para uma extração aprofundada de dados e síntese metodológica.

3. Resultados

Esta seção apresenta a síntese dos resultados obtidos a partir da condução da RSL. A Tabela 4 sumariza os estudos selecionados ao final de todo o processo de triagem, os quais apresentam abordagens alinhadas com o tema investigado.

3.1. Sumarização dos trabalhos

O estudo de [Tian et al. 2026] apresenta um fluxo de trabalho totalmente autônomo baseado em agentes e LLMs, com o objetivo de identificar sinais precoces de declínio cognitivo a partir de prontuários médicos e notas clínicas, eliminando a necessidade de intervenção humana após sua implantação. A arquitetura proposta utiliza o modelo *Llama 3.1 8B* para coordenar a atuação de cinco agentes especializados: um Agente Especialista (*Specialist Agent*) responsável pela classificação primária e raciocínio clínico, apoiado por dois Agentes Melhoradores (*Improver agents*) e dois Agentes Sumarizadores (*Summarizer agents*) focados estritamente na otimização da sensibilidade e da especificidade dos *prompts*. Por meio de um processo iterativo de refinamento, o sistema avalia os seus próprios erros (falsos positivos e falsos negativos) e reescreve automaticamente as instruções até atingir os limiares de desempenho desejados. Segundo [Tian et al. 2026], por meio de avaliação especializada, revelou que o sistema em 44% dos falsos negativos apresenta um raciocínio superior às anotações humanas iniciais, propondo assim, um futuro em que a IA poderá apoiar e facilitar o trabalho clínico.

[Spiess et al. 2025] desenvolveu o AutoPDL, um sistema automatizado para a descoberta de configurações ideais para agentes baseados em LLMs. Esse sistema tem como princípio superar os testes e ajustes manuais da engenharia de *prompts* pois são tediosos

Tabela 4. Artigos selecionados para revisão sistemática.

Autor/Ano	Título	Base de dados
[Tian et al. 2026]	An autonomous agentic workflow for clinical detection of cognitive concerns using large language models	Scopus
[Spiess et al. 2025]	AutoPDL: Automatic Prompt Optimization for LLM Agents	Scopus
[Dorsch et al. 2025]	COMPASS: A Process Mining-based Methodology for Prompt Optimization of Large Language Model Agents	Scopus
[Jalori et al. 2025]	FLAIRR-TS – Forecasting LLM-Agents with Iterative Refinement and Retrieval for Time Series	Scopus
[Hong et al. 2026]	PEEM: Prompt Engineering Evaluation Metrics for Interpretable Joint Evaluation of Prompts and Responses	arXiv
[Purpura et al. 2026]	Enhancing LLM Instruction Following: An Evaluation-Driven Multi-Agentic Workflow for Prompt Instructions Optimization	arXiv

e propensos a erros. Os autores formularam essa estrutura inspirados no *Automated Machine Learning (AutoML)*, combinando padrões de prompt de alto nível (incluindo abordagens não-agênticas como *Zero-Shot* e *Chain-of-Thought*, e padrões agênticos como ReAct e ReWOO) quanto o conteúdo específico do prompt, como o número de demonstrações *few-shot* e as instruções. Um diferencial de sua arquitetura é o uso da *Prompt Declaration Language* (PDL), uma linguagem baseada em YAML (YAML Ain't Markup Language) que garante que o resultado final da otimização seja um programa estruturado, legível por humanos, editável e executável. As avaliações foram conduzidas em três tarefas distintas (perguntas e respostas, matemática e geração de código) e com sete modelos de linguagem variando de 3 a 70 bilhões de parâmetros, demonstraram ganhos consistentes de precisão, alcançando melhorias de até 67,5 pontos percentuais em relação às linhas de base *zero-shot*.

[Dorsch et al. 2025] introduz uma metodologia que utiliza técnicas de Mineração de Processos (*Process Mining*) com a avaliação e otimização de agentes baseados em LLMs. O *framework* COMPASS tem dois objetivos principais: melhorar o desempenho do agente por meio de informações comportamentais e verificar a conformidade com o comportamento especificado. Para isso, o processo possui cinco fases sistemáticas: (1) planejamento do projeto; (2) extração de dados brutos das interações do agente; (3) transformação desses dados em *logs* de eventos (rastreios sequenciais de ações); (4) exploração analítica do comportamento; e, por fim, (5) a provisão de *feedback* acionável por meio de verificações de conformidade ou criação de diretrizes para o redesenho dos *prompts*. Para validar a eficácia dessa arquitetura, conduziu-se um estudo de caso envolvendo Graf-von-Data, que é um agente de IA focado em interrogar bases de dados

complexas usando linguagem natural. Como resultado, COMPASS apresenta melhorias na otimização de *prompts baseada em dados*, além de ganhos de desempenho alcançados por meio diretrizes usando *Process Mining*.

Segundo [Purpura et al. 2026], embora os LLMs sejam excelentes em compreender a tarefa principal e gerar conteúdo semanticamente relevante, eles frequentemente falham em obedecer a regras formais, levando a resultados conceitualmente corretos, mas processualmente falhos. Para mitigar esse problema, o estudo introduz um fluxo de trabalho multiagente que avalia a conformidade de uma resposta com cada restrição especificada. Um diferencial da proposta é a utilização de um sistema orientado a avaliação (*evaluation-driven*), onde as instruções são reescritas de forma iterativa por agentes utilizando pontuações quantitativas de conformidade como *feedback* direto. Os testes empíricos, realizados com modelos de código aberto como Llama 3.1 8B e Mixtral-8x7B, demonstraram que essa abordagem gera *prompts* revisados capazes de alcançar taxas de conformidade significativamente superiores em comparação com técnicas tradicionais de refinamento, evidenciando a eficácia da colaboração entre agentes para a otimização automática de instruções.

O estudo de [Jalori et al. 2025] aborda o desafio de utilizar LLMs para a previsão de séries temporais uma tarefa que exige pré-processamento exaustivo, ajuste fino de pesos (*fine-tuning*) ou uma engenharia de *prompts* excessivamente manual e *ad hoc*[1]. Para solucionar essa lacuna, os autores propõem o FLAIRR-TS, um *framework* de otimização de *prompts* em tempo de inferência (*test-time*) que opera inteiramente sem a atualização de pesos do modelo. A arquitetura coordena três agentes distintos: um Agente Previsor (*Forecaster-agent*), responsável por gerar as previsões numéricas a partir de um prompt inicial; um Agente de Recuperação (*Retrieval agent*), que atua como um sistema de Geração Aumentada por Recuperação (RAG) especializado, buscando segmentos históricos semanticamente semelhantes em uma base de dados para enriquecer o contexto temporal; e por fim, um Agente Refinador (*Refiner agent*). Este último atua como um otimizador com estado (*stateful*), analisando iterativamente a trajetória completa das tentativas de previsão anteriores e seus respectivos erros quantitativos (como o Erro Médio Absoluto, Mean Absolute Error - MAE) para reformular de forma autônoma as instruções do prompt fornecidas ao Agente Previsor. O ciclo de refinamento iterativo continua até que as melhorias na previsão caiam abaixo de um limiar de parada pré-definido. Avaliações foram conduzidas em diversos conjuntos de dados de referência (como ETT, ILINet e clima) utilizando diferentes modelos base (Gemini 2.5 Pro, Gemini 2 Flash e DeepSeek-V3), comprovaram que o FLAIRR-TS melhora substancialmente a precisão das previsões em relação a *prompts* estáticos e métodos RAG não iterativos. Sua principal contribuição não é superar todos os prompts ajustados manualmente, mas sim reduzir o trabalho exaustivo de ajuste manual.

Por fim, [Hong et al. 2026] introduz um novo método para avaliação de LLMs, tanto de instruções quanto de respostas geradas. PEEM é um *framework* que realiza a avaliação conjunta e interpretável, esse sistema estabelece uma rubrica composta por nove eixos de avaliação distintos: três critérios focados exclusivamente no *prompt* (clareza/estrutura, qualidade linguística e justiça/imparcialidade) e seis critérios direcionados

[1]ad hoc é um termo usado para descrever soluções temporárias que não possuem intenção de ser reutilizáveis ou gerais. Normalmente, essas soluções tendem a resolver problemas específicos.

à resposta (acurácia, coerência, relevância, objetividade, clareza e concisão). Utilizando um LLM no papel de avaliador autônomo (*LLM-as-a-judge*), PEEM vai além da simples geração de notas escalares ao produzir justificativas detalhadas em linguagem natural que explicam e fundamentam cada pontuação atribuída. Experimentos conduzidos em sete *benchmarks* confirmaram a acurácia de PEEM, demonstrando alta sensibilidade para detectar manipulações adversariais semânticas, este tipo de manipulação é caracterizado pela elaboração maliciosa de uma instrução com o propósito de contornar os filtros de seguranças e as diretrizes do sistema. Outro aspecto relevante a se destacar desse estudo é a comprovação de que as justificativas e as notas de PEEM podem ser diretamente injetadas como *feedback* em um ciclo iterativo de reescrita de *prompts zero-shot*. Esse fluxo de otimização, guiado puramente pelas avaliações autônomas de PEEM, elevou a precisão final em até 11,7%, superando diversas abordagens de aprendizado por reforço e otimização supervisionada, evidenciando que avaliações estruturadas e interpretáveis são ótimas abordagens para a otimização e o refinamento automatizado de instruções por agentes.

3.2. Síntese dos estudos

A análise dos estudos selecionados revela uma transição clara na literatura: da engenharia de *prompts* manual e estática para arquiteturas autônomas e iterativas guiadas por agentes. Os trabalhos demonstram que a utilização de Sistemas Multiagentes, onde diferentes instâncias assumem papéis especializados, como geração, avaliação e sumarização, permitem otimizar as instruções de forma contínua e sem a necessidade de intervenção humana direta. Essas arquiteturas utilizam os erros estruturais ou as pontuações de tentativas anteriores como *feedback* direto para que agentes refinadores reescrevam os *prompts*, adaptando a solicitação dinamicamente para contornar as limitações de compreensão do modelo.

Além disso, a síntese aponta que o sucesso desse refinamento automatizado depende intrinsecamente de *frameworks* de avaliação integrados ao ciclo iterativo. Mecanismos que avaliam quantitativamente a qualidade da resposta gerada, como a conformidade a restrições e o uso do próprio modelo como juiz (*LLM-as-a-judge*), fornecem o limiar de precisão necessário para guiar as decisões dos agentes. Os estudos evidenciam que, ao atrelar métricas rigorosas e justificativas detalhadas ao processo de reescrita *zero-shot*, os sistemas multiagentes conseguem formular *prompts* significativamente mais completos, resultando em respostas finais de maior acurácia, confiabilidade e alinhamento com a intenção do usuário.

4. Conclusão

A presente RSL evidenciou que a Engenharia de *prompts* está passando por uma transição metodológica fundamental, migrando da experimentação heurística e manual para a automação impulsionada por Sistemas Multiagentes. A investigação demonstrou que a delegação do processo de criação e refinamento de instruções para agentes autônomos resolve um dos maiores gargalos na adoção de LLMs: sua forte dependência da habilidade do usuário em formular instruções iniciais perfeitas.

Os resultados extraídos da literatura confirmam que o êxito dessas arquiteturas modernas reside predominantemente no refinamento iterativo orientado por avaliação.

Trabalhos como [Purpura et al. 2026], comprovam que sistemas que utilizam agentes avaliadores para pontuar respostas e fornecer *feedback* direto aos agentes planejadores conseguem reescrever as instruções de forma autônoma até que os limiares de qualidade e as restrições da tarefa sejam cumpridos. A adoção de *frameworks* de métricas estruturadas e interpretáveis é essencial para esse ciclo, pois permite que o sistema atue como juiz [Hong et al. 2026], gerando justificativas detalhadas que guiam a otimização do prompt de maneira *zero-shot* e sem intervenção humana.

Além disso, a implementação de fluxos de trabalho com múltiplos agentes especializados, provou ser capaz de alcançar e frequentemente superar o desempenho da engenharia de *prompts* realizada por especialistas humanos no domínio clínico [Hong et al. 2026] e de geração de código [Spiess et al. 2025].

De maneira geral, a revisão permitiu identificar diferentes estratégias para a automação da Engenharia de *Prompts*, bem como os principais métodos de avaliação utilizados para mensurar a qualidade das instruções e das respostas geradas. Os achados demonstram que a combinação entre sistemas multiagentes, refinamento iterativo e mecanismos de avaliação são essenciais para o desenvolvimento de soluções capazes de melhorar a interação entre usuários e modelos de linguagem. Além disso, os resultados obtidos fornecem uma base teórica para o desenvolvimento de trabalhos futuros, contribuindo para a compreensão das técnicas, arquiteturas e métricas mais relevantes na construção de agentes especializados em geração e refinamento automatizado de *prompts*.

Referências

Dorsch, R., Henselmann, D., and Harth, A. (2025). Compass: A process mining-based methodology for prompt optimization of large language model agents. *CEUR Workshop Proceedings*, 3996:29 – 37.

Guo, T., Chen, X., Wang, Y., Chang, R., Pei, S., Chawla, N. V., Wiest, O., and Zhang, X. (2024). Large language model based multi-agents: A survey of progress and challenges. *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence Survey Track*, pages 8048–8057.

Hong, M., Lee, E., Park, S., and Kim, J. (2026). Peem: Prompt engineering evaluation metrics for interpretable joint evaluation of prompts and responses in llms. *IEEE Access*, 14:53581–53600.

Jalori, G., Verma, P., and Arık, S. (2025). Flairr-ts – forecasting llm-agents with iterative refinement and retrieval for time series. *EMNLP 2025 - 2025 Conference on Empirical Methods in Natural Language Processing, Findings of EMNLP 2025*, page 15427 – 15437.

Jason Wei, Xuezhi Wang, D. S. M. B. B. I. F. X. E. H. C. Q. V. L. and Zhou, D. (2022). Chain-of-thought prompting elicits reasoning in large language models. *Proceedings of the 36th International Conference on Neural Information Processing Systems (NIPS '22)*, page 24824–24837.

Kitchenham, B. A. and Charters, S. (2007). Guidelines for performing systematic literature reviews in software engineering – version 2.3. Technical report, Keele/Staffs-UK and Durham-UK.

Nakagawa, E. Y., Scannavino, K. R. F., Fabbri, S. C. P. F., and Ferrari, F. C. (2017). *Revisão Sistemática da Literatura em Engenharia de Software: teoria e prática*. Elsevier.

Page, M. J., McKenzie, J. E., Bossuyt, P. M., Boutron, I., Hoffmann, T. C., Mulrow, C. D., Shamseer, L., Tetzlaff, J. M., Akl, E. A., Brennan, S. E., Chou, R., Glanville, J., Grimshaw, J. M., Hróbjartsson, A., Lalu, M. M., Li, T., Loder, E. W., Mayo-Wilson, E., McDonald, S., McGuinness, L. A., Stewart, L. A., Thomas, J., Tricco, A. C., Welch, V. A., Whiting, P., and Moher, D. (2021). The prisma 2020 statement: an updated guideline for reporting systematic reviews. *BMJ*, 372.

Pai, M., McCulloch, M., Gorman, J. D., Pai, N., Enanoria, W., Kennedy, G., Tharyan, P., and Colford Jr., J. M. (2004). Systematic reviews and meta-analyses: An illustrated, step-by-step guide. *National Medical Journal of India*, 17(2):86–95.

Purpura, A., Wang, L., Badyal, S., Beaufrand, E., and Faulkner, A. (2026). Enhancing llm instruction following: An evaluation-driven multi-agentic workflow for prompt instructions optimization.

Sahoo, P., Singh, A. K., Saha, S., Jain, V., Mondal, S., and Chadha, A. (2025). A systematic survey of prompt engineering in large language models: Techniques and applications.

Spiess, C., Vaziri, M., Mandel, L., and Hirzel, M. (2025). Autopdl: Automatic prompt optimization for llm agents. *Proceedings of Machine Learning Research*, 293.

Tian, J., Fard, P., Cagan, C., et al. (2026). An autonomous agentic workflow for clinical detection of cognitive concerns using large language models. *npj Digital Medicine*, 9(1):51.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. (2023). Attention is all you need.

Zhao, W. X., Zhou, K., Li, J., Tang, T., Wang, X., Hou, Y., Min, Y., Zhang, B., Zhang, J., Dong, Z., Du, Y., Yang, C., Chen, Y., Chen, Z., Jiang, J., Ren, R., Li, Y., Tang, X., Liu, Z., Liu, P., Nie, J.-Y., and Wen, J.-R. (2026). A survey of large language models. *Front. Comput. Sci. 20*.