UNIVERSIDADE ESTADUAL PAULISTA "JÚLIO DE MESQUITA FILHO"
FACULDADE DE CIÊNCIAS - CAMPUS BAURU
DEPARTAMENTO DE COMPUTAÇÃO
BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

SOFIA AZEVEDO ROSA

CONSTRUÇÃO DE AGENTE INTELIGENTE PARA RESOLUÇÃO DE QUESTÕES DE VESTIBULAR COM LLM

BAURU
Novembro/2025

SOFIA AZEVEDO ROSA

CONSTRUÇÃO DE AGENTE INTELIGENTE PARA RESOLUÇÃO DE QUESTÕES DE VESTIBULAR COM LLM

Trabalho de Conclusão de Curso do Curso de Ciência da Computação da Universidade Estadual Paulista "Júlio de Mesquita Filho", Faculdade de Ciências, Campus Bauru.
Orientador: Prof. Dr. João Paulo Papa
Coorientador: Prof. Ms. Pedro Henrique Paiola

BAURU
Novembro/2025

R788c

Rosa, Sofia Azevedo

Construção de agente inteligente para a resolução de questões de vestibular com LLM / Sofia Azevedo Rosa. -- Bauru, 2025

63 f. : il., tabs.

Trabalho de conclusão de curso (Bacharelado - Ciência da Computação) - Universidade Estadual Paulista (UNESP), Faculdade de Ciências, Bauru

Orientador: João Paulo Papa

Coorientador: Pedro Henrique Paiola

1. Agentes inteligentes (Software). 2. Processamento de linguagem natural (Computação). 3. Testes educacionais. 4. Sistemas especialistas (Computação). I. Título.

Sistema de geração automática de fichas catalográficas da Unesp. Dados fornecidos pelo autor(a).

Sofia Azevedo Rosa

Construção de Agente Inteligente Para Resolução de Questões de Vestibular com LLM

Trabalho de Conclusão de Curso do Curso de Ciência da Computação da Universidade Estadual Paulista "Júlio de Mesquita Filho", Faculdade de Ciências, Campus Bauru.

Banca Examinadora

Prof. Dr. João Paulo Papa
Orientador
Universidade Estadual Paulista "Júlio de Mesquita Filho"
Faculdade de Ciências
Departamento de Ciência da Computação

Prof. Dra Simone das G Domingues Prado
Universidade Estadual Paulista "Júlio de Mesquita Filho"
Faculdade de Ciências
Departamento de Ciência da Computação

Prof. Dr Leandro Aparecido Passos Junior
Universidade Estadual Paulista "Júlio de Mesquita Filho"
Faculdade de Ciências
Departamento de Ciência da Computação

Bauru, 11 de novembro de 2025.

Para mamãe e papai. Tenho certeza que sem vocês eu não estaria nem perto de onde estou agora.

Agradecimentos

Esse projeto marca a conclusão de um ciclo muito importante na minha jornada de formação e é resultado do esforço de diversas pessoas e, a todas, meu mais sincero "muito obrigada".

Primeiramente, a Deus por todas as bênçãos, por me capacitar e iluminar durante toda a minha formação.

Agradeço aos meus pais por absolutamente tudo: por terem sempre sido a minha base e o lugar seguro do qual eu precisava, sem nunca hesitar em fazer o que era melhor pra mim; por tudo que me ensinaram; pelo amor que, como vocês sempre fizeram questão de demonstrar, é incondicional; pela paciência e carinho; por acreditarem em mim em todo o momento e, finalmente, por sempre colocarem a minha educação como prioridade e assim, tornarem a faculdade e esse projeto possíveis.

Também aos meus avós, que sempre acreditaram em mim e me incentivaram de diversas formas em todas as etapas da minha vida. Vocês também determinaram muito do caminho até aqui, possibilitando muitas das oportunidades que tive o privilégio de ter. Ao Vovô, especialmente: em muitos momentos me peguei pensando que queria te mostrar o resultado de todo seu incentivo, mas tenho certeza que você acompanhou tudo bem de pertinho.

Agradeço a minha tia Lu, pelo apoio de sempre e por me ensinar tanto sobre a vida. Você é um pilar extremamente importante pra minha formação, desde que me deu meu primeiro livro da vida. Também ao meu tio Gui, pelo apoio e carinho desde o nosso primeiro contato.

Agradeço aos meus amigos, que fizeram a faculdade ser uma experiência mais leve e divertida, e também por me ensinarem tanto sobre carinho e amizade no nosso cotidiano. Em especial, agradeço ao Vini, meu namorado, que foi meu parceiro durante todo esse período, sendo parte essencial dessa vivência. Em todos os momentos, os positivos ou negativos, de tranquilidade ou ansiedade, você foi um porto seguro pra mim. A faculdade teria sido muito mais difícil sem sua companhia e apoio constantes. Sou muito grata por ter te conhecido e te ter na minha vida.

Por fim, agradeço a todos os meus professores. Cada um deles me transformou de alguma forma e, claro, me ensinou muito. Não seria a mesma profissional se tivesse sido diferente. Principalmente, agradeço aos professores que me orientaram de forma direta durante a realização desse trabalho, João Paulo Papa e Pedro Henrique Paiola; registro também minha sincera gratidão ao Gabriel Lino, cuja colaboração e apoio no desenvolvimento foram igualmente valiosos. Muito obrigada a todos vocês por toda a paciência e ensinamentos durante esse processo tão difícil.

O verdadeiro problema não é se as máquinas pensam, mas se os homens o fazem.
(B. F. Skinner)

Resumo

Este trabalho investigou o uso de agentes inteligentes baseados em modelos de linguagem de grande escala (LLMs) para a resolução de questões objetivas de vestibulares brasileiros. Foi proposta uma arquitetura em múltiplos passos que integra o modelo a ferramentas externas de cálculo e enciclopédia, e sua performance foi comparada ao uso direto dos LLMs nas mesmas provas (ENEM, USP e Unicamp). Os resultados mostraram que, com modelos de menor capacidade (como Llama 3.1 e Mistral), o agente apresentou desempenho inferior ao uso direto, devido especialmente a falhas na recuperação de informações e à interpretação de saídas intermediárias. No entanto, ao empregar um modelo mais robusto (GPT-4.1 Mini), o agente obteve ganhos expressivos nas questões que exigem raciocínio matemático, evidenciando que a abordagem é promissora em cenários com ferramentas e modelos mais adequados ao domínio. O estudo contribui ao demonstrar experimentalmente os limites e potencial dessa estratégia em contextos educacionais reais.

Palavras-chave: agentes inteligentes; modelos de linguagem; provas de vestibular.

Abstract

This work investigated the use of intelligent agents based on large language models (LLMs) for solving multiple-choice questions from Brazilian university entrance exams. A multi-step architecture was proposed, integrating the model with external tools for calculation and encyclopedic retrieval, and its performance was compared to the direct use of LLMs on the same exams (ENEM, USP, and Unicamp). The results showed that, when using lower-capacity models such as Llama 3.1 and Mistral, the agent performed worse than direct usage, mainly due to failures in information retrieval and interpretation of intermediate outputs. However, when employing a more robust model (GPT-4.1 Mini), the agent achieved significant gains on mathematically demanding questions, indicating that the approach is promising in scenarios supported by more suitable models and tools. This study contributes by experimentally demonstrating both the limitations and the potential of this strategy in real educational contexts.
Keywords: intelligent agents; language models; university entrance exams.

Lista de figuras

Lista de tabelas

Lista de abreviaturas e siglas

LLMs	Large Language Models
SLMs	Small Language Models
PLN	Processamento de Linguagem Natural
API	Application Programming Interface
RAG	Retrieval-Augmented Generation
CoT	Chain-of-Thought
PRK	Prior Knowledge
TU	Text Understanding
IU	Image Understanding
MR	Mathematical Reasoning
ML	Multilingual
BK	Brazilian Knowledge
EK	Encyclopedic Knowledge
IC	Image Comprehension
DS	Domain Specific
CE	Chemistry Elements

Sumário

1 Introdução

Os Grandes Modelos de Linguagem (*Large Language Models* - LLMs), como o GPT e *LLaMa*, têm demonstrado impressionante capacidade de compreensão e geração de textos, revolucionando a área de processamento de linguagem natural (PLN) e se tornando ferramentas úteis para diversas tarefas cotidianas, como atendimento virtual, geração de código e geração de conteúdo publicitário.

A eficiência na compreensão textual desses modelos deriva da arquitetura *Transformer*, cuja estrutura baseada em mecanismos de atenção permite capturar dependências contextuais de maneira paralela e escalável (VASWANI et al., 2023). Por sua vez, a versatilidade e capacidade generativa dos LLMs são impulsionadas pelo treinamento em larga escala, envolvendo uma quantidade muito grande de dados e o uso massivo de poder computacional (WEI et al., 2022).

No entanto, esses modelos ainda enfrentam limitações importantes que impactam a sua confiabilidade: os LLMs podem gerar informações incoerentes, enviesadas ou incorretas, além de apresentarem dificuldades ao lidar com lógica e matemática. Essas limitações se tornam críticas em tarefas específicas que envolvem raciocínio ou informações sensíveis, tornando necessária a integração de mecanismos complementares que ampliem suas capacidades.

Alguns mecanismos já explorados incluem a engenharia de *prompts* e o uso de geração aumentada por recuperação (*Retrieval-Augmented Generation* - RAG). A primeira permite direcionar o comportamento do modelo por meio de instruções em linguagem natural ou da inclusão de exemplos para guiar o modelo, como ocorre na técnica de *few-shot prompting*. Já a segunda incorpora fontes externas de informação durante a geração de respostas, permitindo que o modelo recupere dados relevantes para produzir informações fundamentadas e mais confiáveis (LEWIS et al., 2021).

Além disso, os grandes modelos de linguagem têm sido recentemente explorados como base para o desenvolvimento de agentes inteligentes capazes de realizar tarefas mais complexas de maneira mais precisa. Nessas arquiteturas, os modelos de linguagem funcionam como o núcleo do agente, que pode perceber o ambiente e tomar decisões com base em suas entradas, utilizando ferramentas externas para complementar suas habilidades (XI et al., 2023).

A ideia de agentes inteligentes é tradicional no campo da IA, sempre centrando-se nos conceitos de autonomia e pró-atividade (RUSSELL; NORVIG, 2009). As habilidades textuais dos LLMs permitem a emulação de todas essas competências, o que os torna uma opção viável para ser o componente central desses sistemas. Essa abordagem aproveita o potencial linguístico dos modelos de linguagem e os torna aptos para a realização de tarefas em que o seu raciocínio isolado não seria suficiente.

Diante dessa evolução, este trabalho parte do desafio imposto pela limitação dos grandes modelos de linguagem na resolução de questões multidisciplinares, como as presentes nos vestibulares brasileiros. Apesar de seu potencial, os LLMs isolados nem sempre alcançam desempenho satisfatório, principalmente em questões de matemática (NUNES et al., 2023). Nesse contexto, a construção de um agente inteligente baseado em um grande modelo de linguagem surge como uma solução mais promissora para o problema investigado, o que configura o principal objetivo desta pesquisa.

Ademais, o estudo busca avaliar o desempenho do agente desenvolvido e compará-lo com os resultados obtidos por LLMs utilizados de forma isolada, de modo a analisar os ganhos proporcionados pela abordagem proposta.

Dessa forma, pretende-se não apenas contribuir para o avanço das aplicações dos LLMs em contextos educacionais, como também reforçar o potencial dos agentes inteligentes como estrutura intermediária essencial para o uso eficaz dessas tecnologias em tarefas de alta complexidade.

1.1 Problema

Os LLMs têm se destacado por sua capacidade de geração de texto e resolução de problemas diversos, tornando-se uma ferramenta muito utilizada para tarefas cotidianas e até mesmo em contextos específicos, incluindo aplicações no âmbito educacional. Em especial, destaca-se a possibilidade de utilizar esses modelos na resolução de questões de múltipla escolha presentes em vestibulares brasileiros, como o ENEM (Exame Nacional do Ensino Médio), que exigem conhecimento multidisciplinar e raciocínio interpretativo.

No entanto, apesar de seu grande potencial, os LLMs enfrentam limitações que podem comprometer o desempenho na resolução de questões de vestibular, como a geração de informações incorretas ou fora de contexto e, principalmente, uma notável dificuldade em lidar com lógica e matemática. Uma avaliação, feita por Nunes et al. (2023), indicou que os modelos de linguagem apresentam desempenho inferior em matemática em comparação com outras áreas do conhecimento.

Esse cenário torna evidente a necessidade do desenvolvimento de novas aplicações que aproveitem o potencial dos LLMs e mitiguem as suas limitações quando se trata de tarefas que exigem, além de conhecimento confiável multidisciplinar, habilidades de lógica e matemática. A simples utilização dos modelos de forma direta não é suficiente para resolver as questões de vestibular de forma totalmente satisfatória. Assim, é preciso expandir as capacidades dos LLMs, integrando a eles o acesso a ferramentas e a fluxos controlados de execução.

1.2 Justificativa

A ascensão dos LLMs impulsionou sua utilização em diversas áreas, inclusive na área da educação. Dessa forma, já existem trabalhos que exploram esses modelos de linguagem como ferramentas de apoio ao ensino, aprendizagem e, mais especificamente, na tarefa de resolução de questões de múltipla escolha de conhecimentos gerais, como as presentes nos vestibulares brasileiros.

Apesar do incessável avanço dos LLMs, seu uso isolado em avaliações complexas ainda apresenta obstáculos já mencionados: a alucinação e a dificuldade com a lógica e a matemática. Uma possível forma de mitigar a dificuldade dos LLMs nas questões de matemática foi apresentada por Superbi et al. (2024), que implementou a pesquisa de questões semelhantes para utilizar como guia de resolução.

Entretanto, ao serem utilizados como agentes, os grandes modelos de linguagem passam a contar com um espaço ampliado de percepção e ação, permitindo que atuem como o "cérebro" do sistema, coordenando percepção e ação. Diversas abordagens de agentes como auxiliares de ensino e aprendizagem têm sido propostas e foram recentemente revisadas por Chu et al. (2025), evidenciando o interesse crescente na construção de agentes baseados em LLMs e aplicados no contexto educacional.

Dessa forma, este trabalho visa desenvolver um agente inteligente coordenado por um LLM capaz de resolver questões de vestibulares brasileiros.

1.3 Objetivos

1.3.1 Objetivo Geral

Desenvolver um agente inteligente baseado em um modelo de linguagem que seja capaz de responder questões de vestibular com maior acurácia

1.3.2 Objetivos Específicos

- Analisar o desempenho de diferentes LLMs em tarefas de compreensão e resolução de questões objetivas e discursivas, para identificar as principais limitações que eles apresentam nessa tarefa
- Utilizar ferramentas que mitiguem as limitações dos modelos de linguagem na tarefa de resolução de questões de vestibular
- Projetar e implementar o fluxo de funcionamento do agente inteligente, gerenciado por um modelo de linguagem, definindo as etapas envolvidas na resolução de uma questão de vestibular e integrando as ferramentas desenvolvidas para cada uma delas

- Avaliar o agente inteligente com a utilização do conjunto de dados *BLUEX* (ALMEIDA et al., 2023), que contém questões de diversos vestibulares brasileiros.

2 Fundamentação Teórica

Neste capítulo, são descritos os principais conceitos, técnicas e trabalhos que embasam o desenvolvimento deste trabalho. Primeiramente, discutem-se os avanços e limitações dos LLMs, além de técnicas auxiliares que buscam aprimorar a utilização desses modelos. Em seguida, são apresentados diversos conceitos fundamentais de agentes inteligentes e também a sua intersecção com os LLMs. Por fim, são analisados trabalhos recentes que também abordam a resolução de questões multidisciplinares e que inspiram o desenvolvimento deste trabalho.

2.1 Desenvolvimento, Avanços e Limitações dos LLMs

Modelos de linguagem são sistemas treinados para prever a próxima palavra em uma sequência de texto, sendo amplamente utilizados em tarefas de compreensão e geração de conteúdo em linguagem natural. Esses modelos são capazes de identificar relações complexas entre as palavras e entender significados subjetivos dependentes do contexto.

Atualmente, esses modelos são majoritariamente construídos seguindo a arquitetura Transformer. Diferentemente de arquiteturas anteriores, os Transformers utilizam o mecanismo de autoatenção, que permite ao modelo ponderar a importância de outras palavras em uma sequência ao processar cada palavra. Essa abordagem possibilita capturar relações de dependência entre palavras distantes (VASWANI et al., 2023).

O funcionamento central de um Transformer baseia-se em duas componentes principais: o *Encoder* (codificador) e o *Decoder* (decodificador). O *Encoder* recebe a sequência de entrada e a transforma em representações vetoriais que capturam o significado de cada palavra com base na influência das demais. Para isso, ele utiliza o mecanismo de autoatenção que permite que o modelo avalie a importância de todas as palavras da sequência para entender cada termo individualmente. O *Decoder*, por sua vez, utiliza essas representações para prever de forma sequencial cada termo da sequência de saída. Esse componente utiliza do mecanismo de autoatenção para garantir que a previsão de cada palavra considere o contexto das palavras geradas anteriormente. Essa estrutura possibilita ao modelo lidar eficientemente com dependências de longo alcance e permite o treinamento paralelo, aumentando tanto a precisão quanto a eficiência do processamento de texto (VASWANI et al., 2023).

Com o sucesso da arquitetura Transformer, a evolução dos modelos de linguagem tem sido marcada por um crescimento exponencial em escala, tanto em termos de número de parâmetros quanto na quantidade de dados de treinamento utilizados. Modelos precursores contavam com milhões de parâmetros, enquanto os LLMs mais recentes possuem bilhões deles. Esses parâmetros são os valores numéricos ajustáveis dentro das conexões neurais da rede que,

no conjunto, formam a memória e o conhecimento internalizado do modelo sobre a linguagem e o mundo.

A partir dessa evolução, surge uma discussão ainda em aberto na literatura sobre a diferenciação dos modelos de pequena (*Small Language Models* - SLMs) e larga escala a partir de sua quantidade de parâmetros. Existem abordagens mais abrangentes, que classificam modelos com a partir de 100 milhões de parâmetros como LLMs (ÖRPEK; TURAL; DESTAN, 2024), enquanto outras reservam o termo para modelos bem maiores, com dezenas de bilhões de parâmetros (JOVANOVIC; CAMPBELL, 2024).

Na prática, porém, a distinção de escala tem sido observada principalmente pelo surgimento de algumas capacidades emergentes nos LLMs, entendida como as habilidades que são possíveis somente a partir de um certo nível de complexidade desses modelos. Entre essas habilidades estão a interpretação de perguntas mais complexas, o desenvolvimento de raciocínios de múltiplos passos e, principalmente, capacidades generalistas, isto é, de realizar tarefas sem treinamento específico adicional (WEI et al., 2022).

Por isso, neste trabalho, adota-se a notação de LLMs para modelos com a partir de 7 bilhões de parâmetros, uma vez que esses já demonstram de forma consistente tais capacidades generalistas e raciocínios sofisticados, mesmo sem *fine-tuning* dedicado para cada tarefa.

Apesar de seu desempenho notável, os modelos de linguagem de larga escala ainda apresentam limitações significativas que podem comprometer sua adoção em diversos contextos sensíveis. Uma das principais limitações é o fenômeno conhecido como alucinação, no qual os LLMs geram informações imprecisas, ainda que expressas com alto grau de confiança. Esse comportamento prejudica seu desempenho em tarefas que exigem precisão e confiabilidade, além de levantar sérias preocupações éticas quanto ao uso responsável da tecnologia.

A revisão de Zhang et al. (2023) classifica o fenômeno da alucinação em três tipos distintos, que podem ocorrer na resposta de um LLM: a inconsistência com a entrada do usuário, quando a resposta não corresponde às instruções fornecidas; a inconsistência com o contexto, que ocorre quando o modelo gera informações que contradizem elementos previamente apresentados na conversa ou no documento; e a inconsistência factual, caracterizada por afirmações incorretas em relação a dados reais. A figura 1 exemplifica esses três casos de alucinação, permitindo uma compreensão visual de como eles se manifestam nas respostas dos modelos.

Diferentes fatores explicam a origem dessa problemática, sendo a natureza e a magnitude dos dados de treinamento dos LLMs as principais causas. Esses modelos são expostos a uma grande quantidade de dados, geralmente coletados automaticamente da internet, sem revisão ou anotação humana. Assim, parte desses dados pode apresentar certos vieses e defasagens, que são internalizados pelos modelos, o que pode gerar inconsistências em suas respostas (ZHANG et al., 2023).

Outra limitação intrínseca aos LLMs é a sua dificuldade em interpretar e resolver

Figura 1 – Exemplos dos tipos de Alucinação

Fonte: Adaptada de Zhang et al. (2023)

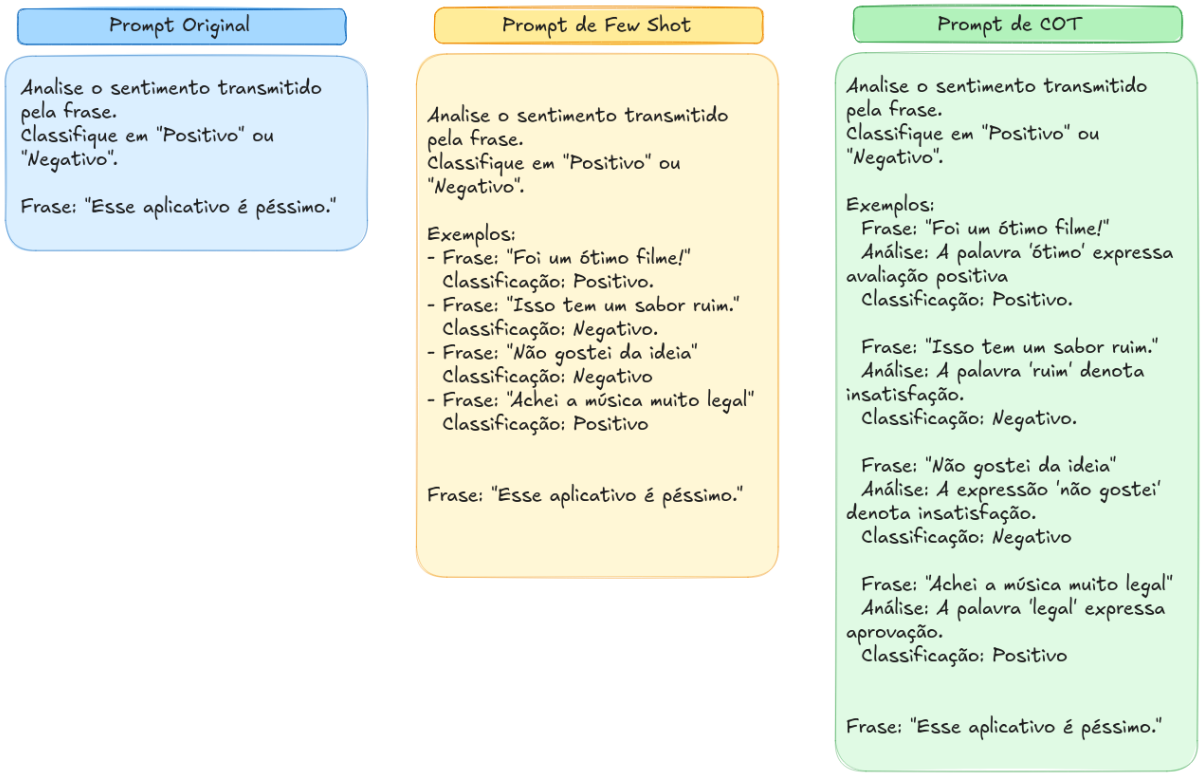
problemas que envolvem raciocínio matemático. Por serem fundamentalmente modelos estatísticos, treinados para prever a próxima palavra em uma sequência com base em padrões de texto, os LLMs não possuem uma capacidade natural de cálculo preciso ou de raciocínio lógico estruturado. Em vez de realizar cálculos como um processador, por exemplo, eles tentam simular a sequência de palavras que corresponde a uma solução correta, baseando-se nas probabilidades aprendidas durante o treinamento. Além dessa dificuldade estrutural dos modelos, os dados utilizados no seu pré treinamento podem afetar sua capacidade de raciocínio matemático: se o *corpus* de treinamento não for suficientemente rico em código ou notação matemática, as habilidades aritméticas do modelo serão prejudicadas.(AHN et al., 2024)

Em vista das limitações inerentes aos grandes modelos de linguagem, a busca por estratégias de mitigação tornou-se essencial para viabilizar o uso dessa tecnologia em diversos contextos. Entre as abordagens mais adotadas, destacam-se as técnicas de engenharia de *prompt*, que englobam o *few-shot prompting* e o *Chain-of-Thought* (COT), o RAG e a integração desses modelos com ferramentas externas.

As técnicas de engenharia de *prompt* permitem que os modelos sejam capacitados para a realização de determinadas tarefas no momento da inferência, sendo consideradas técnicas de aprendizado contextual (*In-Context Learning*). Dentre elas, a abordagem mais notável é o *few-shot prompting*, que basicamente consiste em fornecer ao modelo de linguagem, junto à descrição da tarefa, alguns pares de contexto-resposta que exemplifiquem sua execução. Dessa forma, o modelo utiliza esses exemplos para orientar a geração da saída apropriada para cada entrada, imitando a habilidade humana de aprender a partir de algumas amostras (BROWN et al., 2020).

Avançando no escopo do *few-shot prompting*, a técnica de COT, ou instrução por cadeia de pensamento, introduz um refinamento crucial: em vez de fornecer ao modelo apenas os exemplos de entrada e saída, o *prompt* também inclui uma sequência de etapas de raciocínio intermediárias e explícitas que levam à solução. Isso permite que os LLMs aprendam a decompor problemas complexos de raciocínio em passos mais simples e executáveis (WEI et al., 2023). A figura 2 exemplifica e compara as técnicas de *prompt*.

Figura 2 – Exemplos de tipos de *prompt*



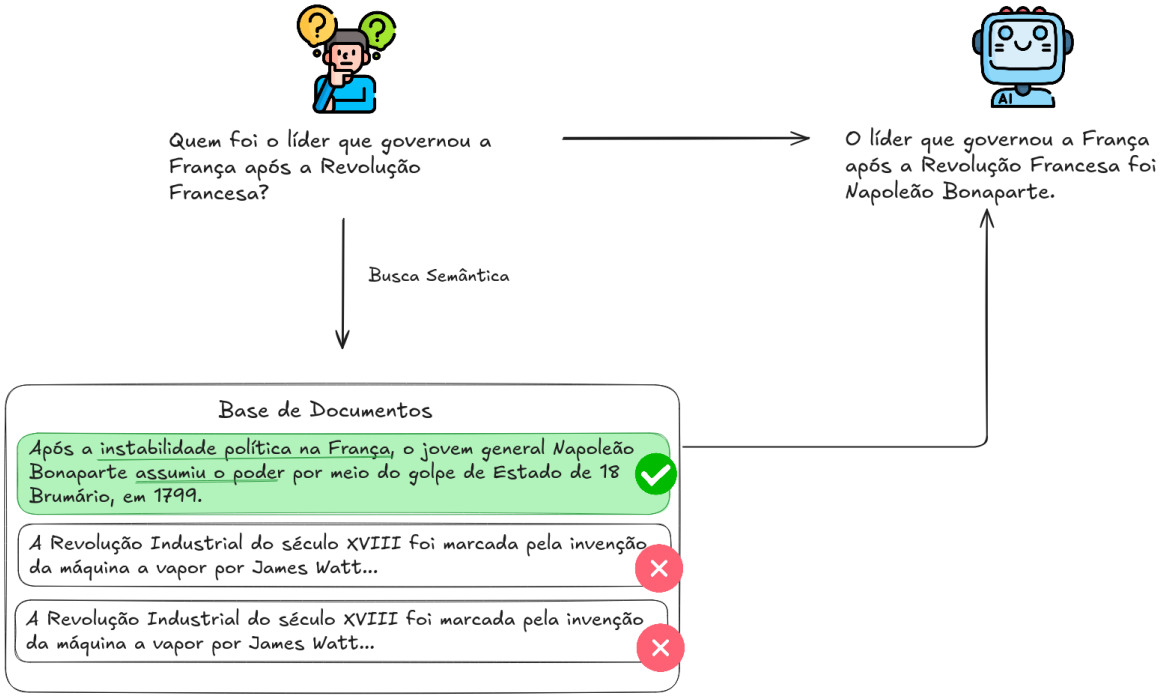
Fonte: Elaborada pelo autor

Outro mecanismo essencial para mitigar as limitações dos LLMs, a técnica de RAG, introduzida por Lewis et al. (2021), visa enriquecer a geração de texto dos LLMs com a recuperação de informação externa, permitindo que as respostas produzidas pelos modelos sejam fundamentadas em dados verificáveis e, consequentemente, mais precisas e confiáveis. O RAG tornou-se essencial para diversas aplicações de LLMs, principalmente aquelas de escopo específico, como Medicina e Direito.

O RAG começa pela recuperação de documentos de uma base de dados e, para isso, são utilizadas técnicas de busca semântica, visando a recuperação de informações relevantes, ou seja, conceitualmente próximas à consulta. Em seguida, o conteúdo desses documentos é passado como contexto para o LLM gerar sua resposta, fundamentando suas inferências nessas informações. Esse processo está representado visualmente na figura 3. Dessa forma, o conteúdo gerado pelo modelo é menos dependente de seu conhecimento interno, o que aumenta a precisão e a confiabilidade de suas respostas. Além disso, a técnica evoluiu para incluir abordagens mais sofisticadas, como pré-filtragem de documentos, validação do conteúdo recuperado e recuperação iterativa, que permitem refinar continuamente as informações fornecidas ao modelo (GAO et al., 2024).

Por fim, a utilização de ferramentas externas pelos modelos de linguagem também representa uma estratégia para minimizar suas limitações. Essa integração permite aos modelos

Figura 3 – Processo do RAG



Fonte: Elaborada pelo Autor

expandir sua capacidade de ação, delegando tarefas específicas e críticas, como a resolução de cálculos matemáticos, a geração de código ou a consulta a bancos de dados em tempo real, para módulos especializados.

Essa abordagem, contudo, é mais complexa do que as estratégias de *prompting* ou RAG tratadas anteriormente. A integração eficaz de ferramentas exige que o LLM não apenas compreenda a intenção do usuário e decomponha tarefas complexas em etapas executáveis, mas também desenvolva a habilidade crucial de entender a funcionalidade e o uso apropriado de cada ferramenta disponível. Por essa razão, a pesquisa atual se concentra em desenvolver estratégias eficazes para capacitar os LLMs a manejar essas ferramentas, utilizando métodos como o ajuste fino (*fine-tuning*) e técnicas avançadas de *prompting* (SHEN, 2024).

Em resumo, os grandes modelos de linguagem marcaram um avanço significativo na inteligência artificial ao ampliar a capacidade de compreensão e geração de texto em linguagem natural. Ainda assim, suas limitações evidenciam a necessidade de abordagens complementares, como técnicas de prompting, recuperação de informações e integração com ferramentas externas. Esses avanços abrem caminho para uma nova etapa de pesquisa e aplicação, centrada em agentes inteligentes baseados em LLM, capazes de raciocinar, planejar e agir de forma mais autônoma.

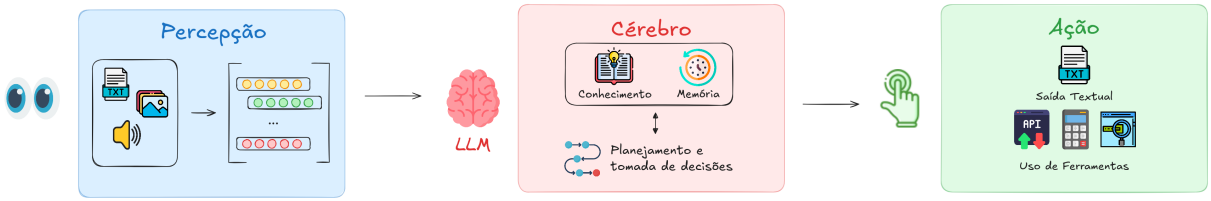
2.2 Agentes Inteligentes e Agentes Baseados em LLM

O conceito de agente inteligente é amplamente reconhecido como um dos fundamentos da área de Inteligência Artificial (IA). Trata-se de uma entidade capaz de perceber seu ambiente por meio de sensores e atuar sobre ele por meio de atuadores (RUSSELL; NORVIG, 2009). Complementarmente, de acordo com Wooldridge e Jennings (1995), um agente inteligente se distingue por quatro características principais: autonomia, reatividade, pró-atividade e habilidade social.

Nesse contexto, os LLMs são considerados um novo paradigma para a construção de agentes inteligentes. Sua habilidade de gerar e interpretar textos possibilita que eles atuem como componente cognitivo central desses agentes, expandindo seu espaço de percepção e ação por meio de percepção multimodal e utilização de ferramentas externas. Dessa forma, os LLMs consolidam-se como uma base promissora para agentes capazes de operar de maneira cada vez mais autônoma e adaptável.

A estrutura conceitual proposta por Xi et al. (2023) para a construção desses agentes estrutura-se em torno de três componentes essenciais: cérebro, percepção e ação. O modelo está representado na figura 4.

Figura 4 – Estrutura conceitual de agentes inteligentes baseados em LLMs



Fonte: Adaptada de Xi et al. (2023)

O Cérebro atua como o núcleo cognitivo do agente e é tipicamente representado pelo próprio LLM. Sua função essencial é o processamento da informação recebida (percepção), a tomada de decisões, o raciocínio e o planejamento da ação. Para realizar essas tarefas, o cérebro se baseia no vasto conhecimento natural do modelo de linguagem e é complementado por um módulo de Memória. A Memória registra o histórico de observações, pensamentos e ações passadas, o que é crucial para a formulação de estratégias coerentes e para lidar com tarefas sequenciais.

O módulo de Percepção do agente é responsável por receber e processar as informações do ambiente. Sua estrutura fundamental lida com a entrada de dados em formato textual. No entanto, para aumentar a capacidade do agente em interagir com o mundo, esse módulo pode ser expandido para incorporar a análise de dados multimodais, como informações visuais ou sonoras. Essa expansão permite que o agente contextualize e compreenda o ambiente de forma

mais abrangente, enriquecendo a informação textual que é passada ao cérebro para a tomada de decisões.

Por fim, o módulo de Ação é o meio pelo qual o agente executa suas decisões no ambiente. A forma de ação mais básica e inerente ao LLM é a geração de texto. Contudo, para estender a capacidade de atuação do agente para além da resposta textual, o módulo de Ação pode ser configurado para permitir o uso de ferramentas. Este mecanismo permite que o agente utilize ferramentas externas, como pesquisa na *internet*, calculadoras ou execução de código, aumentando sua especialização, confiabilidade e eficácia na resolução de problemas práticos.

Em síntese, os agentes inteligentes baseados em LLMs representam uma etapa significativa na evolução da Inteligência Artificial, ao integrarem raciocínio linguístico, percepção multimodal e capacidade de ação autônoma em uma única estrutura coerente. A combinação entre módulos especializados — percepção, cognição, memória e ação — permite que esses agentes apresentem comportamentos mais contextuais, adaptativos e interativos, aproximando-se de uma forma de inteligência artificial mais geral.

2.3 Trabalhos Relacionados

A arquitetura de agentes inteligentes baseados em LLMs, com suas capacidades de raciocínio, memória e uso de ferramentas, oferece uma estratégia para enfrentar problemas complexos do mundo real. Um domínio de aplicação particularmente desafiador é a resolução de questões acadêmicas multidisciplinares, como as encontradas em vestibulares e exames de larga escala. Essas tarefas testam diretamente as limitações dos LLMs, como o raciocínio matemático e a consistência factual, e demandam uma integração sofisticada de habilidades que vai além da simples geração de texto.

Dessa forma, nos últimos anos, diversos estudos têm explorado o uso de agentes baseados em modelos de linguagem para a resolução de problemas complexos em contextos educacionais, especialmente em tarefas nas quais o uso isolado de LLMs não é suficiente para alcançar alta acurácia. No contexto brasileiro, o trabalho de Nunes et al. (2023) avaliou o desempenho dos modelos GPT-3.5 e GPT-4 na resolução de questões do ENEM, aplicando diferentes estratégias de *prompting*. Os resultados revelaram que, embora a técnica de CoT tenha elevado a acurácia para $94,59\%$ na área de Ciências Humanas, o desempenho em Matemática permaneceu limitado, não ultrapassando $72,73\%$. A média máxima geral alcançada na prova de 2022 foi de $87,9\%$, evidenciando que, mesmo com estratégias de engenharia de *prompt*, os LLMs ainda apresentam limitações relevantes — principalmente em domínios que exigem raciocínio estruturado e operações formais.

Buscando superar essas limitações, o trabalho de Superbi et al. (2024) propõe uma abordagem baseada em RAG para melhorar o desempenho dos LLMs em questões de matemática

no ENEM. O sistema utiliza um banco de dados com questões resolvidas, do qual são recuperadas questões semelhantes à pergunta de entrada, oferecendo uma base de raciocínio que orienta o modelo na resolução.

Complementarmente, outras abordagens propõem arquiteturas de agentes para lidar com questões complexas de áreas específicas. O *MathChat* (WU et al., 2024) é um trabalho no qual a resolução de questões complexas de matemática é feita por um sistema multiagente, composto por um agente baseado em LLM, que desenvolve o raciocínio e gera subtarefas, que são executadas por um segundo agente. De forma análoga, o *PhysicsReasoner* (PANG et al., 2024) é um agente que resolve questões complexas de física seguindo um determinado fluxo: extração de variáveis, recuperação de fórmulas e construção de um código que, ao ser executado, retornará a resposta final.

Esses trabalhos evidenciam tanto os avanços quanto as limitações atuais dos LLMs em contextos educacionais. Eles também motivam diretamente a proposta deste trabalho: o desenvolvimento de um agente inteligente voltado especificamente para a resolução de questões de vestibulares brasileiros, com ênfase no ENEM, integrando modelos de linguagem com mecanismos de memória, recuperação de conhecimento e uso de ferramentas externas.

3 Metodologia

Este capítulo descreve os procedimentos metodológicos adotados para o desenvolvimento e avaliação do agente baseado em LLM aplicado à resolução de questões de vestibular.

Inicialmente, serão apresentados os modelos de linguagem e as bases de dados utilizadas no desenvolvimento e na experimentação do agente, seguidos pelas arquiteturas e fluxos explorados para a criação de um agente eficaz na tarefa proposta. Além disso, serão explicados os procedimentos de avaliação quantitativa e qualitativa, que envolvem a comparação entre fluxos de agentes e modelos de linguagem.

3.1 Modelos de Linguagem e Dados

3.1.1 Modelos de Linguagem

Para fundamentar o agente aplicado à resolução de questões de vestibular, foram escolhidos dois LLMs *open-source*: Llama 3.1 (GRATTAFIORI et al., 2024) e Mistral 7B (JIANG et al., 2023). A escolha desses modelos baseia-se em sua acessibilidade, tanto no aspecto financeiro quanto no uso de recursos computacionais. Esses modelos são gratuitos e possuem arquiteturas relativamente leves: embora existam modelos de linguagem com centenas de bilhões de parâmetros, os modelos escolhidos têm entre 7 e 8 bilhões de parâmetros, uma quantidade considerável para a geração e compreensão de texto, mas que, ainda assim, não torna os modelos excessivamente pesados, permitindo sua utilização em contextos com infraestrutura restrita.

Esses modelos representam algumas arquiteturas distintas, desenvolvidas com diferentes focos. O Llama 3.1 é estruturado como um modelo transformer tradicional, visando estabilidade no treinamento para aumento de escala nos parâmetros, o que possibilitou, inclusive, sua versão de 405 bilhões de parâmetros (GRATTAFIORI et al., 2024).

O Mistral, por sua vez, apresenta uma arquitetura sutilmente modificada por algumas escolhas, como, por exemplo, a janela de contexto deslizante (do inglês *Sliding Window Attention*), que possibilita a compreensão de textos mais longos (JIANG et al., 2023).

Em resumo, os modelos foram escolhidos devido à combinação de acessibilidade e eficiência computacional. Embora compartilhem a característica de possuir uma quantidade relativamente modesta de parâmetros, cada um apresenta características distintas em suas arquiteturas, o que proporciona uma base sólida para o desenvolvimento do agente proposto e também para possíveis discussões e comparações entre suas capacidades para a tarefa de responder a perguntas de vestibular.

3.1.2 Bases de Dados

Todo o processo de teste e avaliação do agente baseia-se em questões objetivas de vestibulares brasileiros. Para isso, foram utilizados os conjuntos de dados *BLUEX* (ALMEIDA et al., 2023) e *Enem Challenge* (SILVEIRA; MAUÁ, 2017).

O *BLUEX* reúne questões dos vestibulares da USP e da Unicamp entre 2018 à 2024, acompanhadas de metadados que possibilitam analíses detalhadas. A figura 5 demonstra um exemplo de entrada que inclui, além do enunciado, alternativas e gabaritos, as seguinte categorias anotadas:

- *Subject*: área(s) do conhecimento abordada(s);
- *Prior Knowledge* (PRK): necessidade de conhecimento prévio além do fornecido;
- *Text Understanding* (TU): interpretação de texto;
- *Image Understanding* (IU): interpretação de imagem(s);
- *Mathematical Reasoning* (MR): realização de cálculos;
- *Multilingual* (ML): presença de múltiplas línguas;
- **Brazilian Knowledge** (BK): conteúdos específicos do contexto brasileiro.

Figura 5 – Exemplo de questão no *dataset BLUEX* (ALMEIDA et al., 2023)

Question	O ano de 2017 marca o trigésimo aniversário de um grave acidente de contaminação radioativa, ocorrido em Goiânia em 1987. Na ocasião, uma fonte radioativa, utilizada em um equipamento de radioterapia, foi retirada do prédio abandonado de um hospital e, posteriormente, aberta no ferro velho para onde fora levada. O brilho azulado do pó de césio-137 fascinou o dono do ferro-velho, que compartilhou porções do material altamente radioativo com sua família e amigos, o que teve consequências trágicas. O tempo necessário para que metade da quantidade de césio-137 existente em uma fonte se transforme no elemento não radioativo bário-137 é trinta anos. Em relação a 1987, a fração de césio-137, em %, que existirá na fonte radioativa 120 anos após o acidente, será,aproximadamente,
Alternatives	a) 3,1. b) 6,3. c) 12,5. d) 25,0. e) 50,0.
Answer	B
Subject	chemistry, physics
TU	true
IU	false
MR	true
ML	false
BK	false
BK	true

Fonte: Elaborada pelo autor.

O *ENEM Challenge*, por sua vez, reúne questões do ENEM entre 2009 e 2017, também com anotações que qualificam as habilidades exigidas. A Figura 6 apresenta sua estrutura, que inclui categorias como:

- *Encyclopedic Knowledge* (EK): conhecimento além do cabeçalho;
- *Image Comprehension* (IC): interpretação de imagens;
- *Text Comprehension* (TC): interpretação textual ou senso comum;
- *Domain Specific* (DS): conteúdo específico de uma área;
- *Mathematical Reasoning* (MR): raciocínio matemático;
- *Chemistry Elements* (CE): presença de elementos químicos no enunciado.

Figura 6 – Exemplo de questão no *dataset ENEM Challenge* (SILVEIRA; MAUÁ, 2017)

Header	O fisiologista francês Jean Poiseuille estabeleceu, na primeira metade do século XIX, que o fluxo de sangue por meio de um vaso sanguíneo em uma pessoa é diretamente proporcional à quarta potência da medida do raio desse vaso. Suponha que um médico, efetuando uma angioplastia, aumentou em 10% o raio de um vaso sanguíneo de seu paciente.
Statement	O aumento percentual esperado do fluxo por esse vaso está entre:
Alternatives	a) 7% e 8% b) 9% e 11% c) 20% e 22% d) 39% e 41% e) 46% e 47%
Answer	E
@CE	No
@DS	No
@EK	No
@IC	No
@MR	Yes
@TC	No

Fonte: Elaborada pelo autor

Ambos os conjuntos foram concebidos para a avaliação sistemática de modelos de linguagem, oferecendo anotações que permitem investigar não apenas a taxa de acerto, mas também o tipo de habilidade envolvida em cada questão — o que é essencial para a análise do agente proposto.

Como este trabalho compara modelos com capacidades distintas, especialmente em relação a entrada multimodal, foram selecionadas apenas questões compostas exclusivamente por texto. Essa filtragem garante condições uniformes de processamento e uma avaliação justa da performance entre os modelos considerados.

3.2 Ferramentas e Arquitetura do Agente

3.2.1 Ferramentas

A seguir são descritas as ferramentas externas utilizadas, *Wolfram alpha* e *Wikipédia*, para cálculo e recuperação de informações, respectivamente. Ambas as ferramentas são serviços de interface de programação de aplicações (*Application Programming Interface* - API) de uso gratuito, disponibilizado mediante cadastro prévio.

3.2.1.1 Wolfram Alpha

O Wolfram Alpha é um serviço de conhecimento computacional (do inglês *computational knowledge engine*) que possui a capacidade de responder à perguntas e realizar cálculos complexos. Ele se diferencia de ferramentas de busca, como o Google, por oferecer respostas diretamente a partir do processamento de dados provenientes de diversas fontes confiáveis, ao invés de apenas listar resultados e suas fontes.

Além disso, o Wolfram Alpha disponibiliza uma série de APIs que permitem interações diretas com seu motor de conhecimento. Essas APIs variam em função do tipo de resposta fornecida. Para facilitar a interação do LLM com o serviço, foi escolhida a API que oferece respostas textuais e curtas. O Wolfram Alpha será utilizado exclusivamente para a ferramenta de calculadora, por sua capacidade de interpretação de expressões em texto.

3.2.1.2 Wikipédia

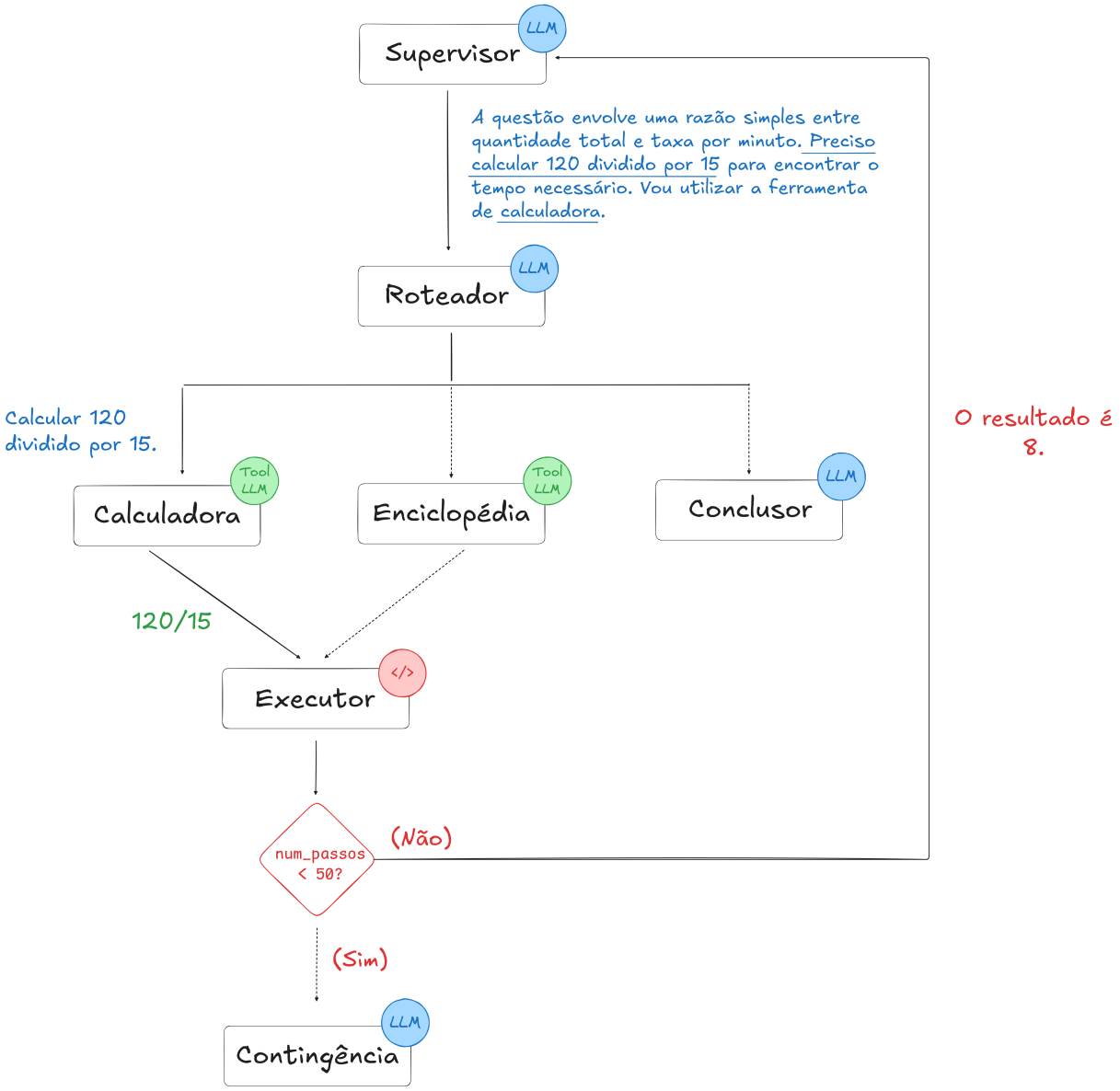
A *Wikipédia* é um projeto *open-source* e colaborativo de enciclopédia *online*, que permite a qualquer usuário criar, editar, corrigir e atualizar artigos sobre uma vasta gama de tópicos. Com mais de 61 milhões de artigos, sendo aproximadamente 1 milhão deles na língua portuguesa, ela se configura como uma fonte dinâmica para consulta de conteúdos de disciplinas escolares e de atualidades, ou seja, de temas abordados nas questões do vestibular.

Da mesma forma que o Wolfram Alpha, a *Wikipédia* também oferece um API para acessar os seus artigos por meio de uma entrada do usuário indicando o assunto. Assim, será oferecida ao agente uma ferramenta baseada na enciclopédia, utilizada para consulta dos conteúdos das questões.

3.2.2 Arquitetura do Agente

A estrutura do agente, representada na figura 7, foi organizada como um fluxo composto por múltiplos nós interligados, cada um com uma responsabilidade específica dentro do processo de resolução da pergunta. Em vez de responder diretamente à entrada do usuário, o agente percorre etapas internas de decisão e, quando necessário, consulta ferramentas externas de forma controlada e explícita.

Figura 7 – Esquemático da arquitetura do agente



Fonte: Desenvolvida pelo autor

A pergunta é inicialmente enviada ao nó supervisor, que atua como coordenador central da resolução. Esse componente interpreta a entrada do usuário e decide, de forma iterativa, quais ações serão realizadas ao longo do processo: se uma ferramenta externa deve ser acionada, como ela será utilizada ou se já existem informações suficientes para produzir uma resposta. A decisão tomada é então repassada ao nó seguinte, o roteador.

O roteador tem o papel exclusivo de interpretar essa decisão e direcionar o fluxo para o estado apropriado. Caso o supervisor tenha optado pelo uso de uma ferramenta, o roteador encaminha a execução para o nó especializado na ferramenta correspondente, acompanhada de uma síntese da instrução do coordenador, que serve como orientação para esse nó durante a execução. Se a resposta já puder ser gerada, ele direciona o fluxo para o nó responsável por estruturar a resposta final.

Existe um nó especializado para cada ferramenta: calculadora (Wolfram Alpha) e enciclopédia (Wikipedia). Essa escolha está alinhada ao princípio de que o coordenador é o único responsável por decidir se e qual ferramenta será utilizada, já que mantém acesso ao contexto completo da pergunta.

Os nós especializados têm como função preparar a entrada adequada para a ferramenta escolhida, transformando a instrução recebida em uma expressão numérica (no caso da calculadora) ou em um termo de busca (no caso da Wikipedia). Essa entrada é então encaminhada ao nó responsável por invocar a ferramenta e receber a sua resposta, que será, posteriormente, levada de volta ao coordenador do fluxo.

Antes de retornar ao coordenador, o fluxo passa por uma etapa de checagem do número de passos já realizados na resolução. Por se tratar de um processo iterativo conduzido por um modelo de linguagem, existe o risco de chamadas redundantes às ferramentas ou de encadeamentos excessivamente longos de raciocínio, o que pode tornar a resolução impraticável. Por isso, verifica-se se o fluxo já atingiu 50 etapas, contabilizando as interações do coordenador, do roteador e todas as chamadas e preparações de ferramentas até o momento. Caso esse limite seja alcançado, a execução é desviada para um nó de encerramento forçado, que recebe a pergunta e as alternativas e produz a resposta final em um único passo, atuando como um LLM independente.

Enquanto o limite não é atingido, o fluxo retorna ao coordenador, que recebe novamente a pergunta original, as alternativas, suas mensagens anteriores e as respostas das ferramentas utilizadas até aquele ponto. A partir desse histórico, o coordenador decide o próximo passo e continua o raciocínio. Essa estrutura iterativa é intencional: ela força a decomposição da resolução em etapas menores e controladas, permitindo que o agente recorra a ferramentas externas apenas quando necessário, em vez de tentar resolver a questão de uma só vez.

3.3 Procedimentos de avaliação

Para avaliar a eficácia dos modelos e dos fluxos de agentes propostos, serão realizados testes com provas completas das universidades USP, Unicamp e Enem. Ambos os tipos de avaliação, tanto quantitativa quanto qualitativa, serão aplicados para medir o desempenho do agente de maneira abrangente. A avaliação quantitativa se concentrará na acurácia, ou seja, no número de questões corretamente respondidas, proporcionando uma medida objetiva da performance do agente em comparação com o padrão de respostas esperadas.

Além disso, a análise qualitativa buscará entender como o agente utiliza as ferramentas disponíveis e como organiza o fluxo de raciocínio ao abordar as questões. Serão analisados aspectos como a lógica do fluxo de trabalho do agente, a autonomia do modelo em cada etapa do processo e a coerência das respostas geradas.

3.3.1 Análise quantitativa

A avaliação quantitativa será realizada com base nos *datasets* utilizados, que apresentam as questões anotadas, permitindo uma análise detalhada da performance do agente. A primeira abordagem quantitativa considerará a acurácia geral, ou seja, o número total de questões corretamente respondidas pelo agente, em comparação com o número de questões disponíveis em cada prova. Além disso, a análise será segmentada por prova, calculando a acurácia por prova específica (USP, Unicamp e Enem), proporcionando uma visão mais detalhada do desempenho do agente em diferentes contextos e dificuldades.

Além da acurácia geral, a avaliação quantitativa será aprofundada por meio das anotações associadas às questões. Serão analisadas questões em termos de sua matéria relacionada, como Ciências Humanas, Ciências da Natureza, Linguagens ou Matemática, permitindo verificar como o agente se sai em diferentes áreas do conhecimento. Além disso, será avaliada a capacidade do agente em lidar com questões que exigem raciocínio matemático e interpretação de texto. Esse nível de detalhamento permitirá uma análise mais refinada da performance do agente, identificando áreas de força e oportunidades de melhoria em seu desempenho.

3.3.2 Análise qualitativa

A análise qualitativa será realizada com o objetivo de examinar mais profundamente o comportamento do agente durante a execução de suas respostas, considerando aspectos como o uso das ferramentas e a autonomia do modelo na seleção dessas ferramentas. Em vez de focar apenas nos resultados numéricos, a análise qualitativa busca entender como o agente organiza seu raciocínio, como ele utiliza as ferramentas disponíveis e qual é o papel da autonomia na geração das respostas. Serão observadas as execuções dos fluxos do agente para avaliar como

ele escolhe e aplica as ferramentas de forma dinâmica, dependendo da complexidade e dos requisitos da questão.

Essa análise tem como objetivo identificar padrões e características positivas ou negativas no comportamento do agente ao seguir o fluxo proposto. No entanto, trata-se de uma avaliação conduzida sobre um conjunto limitado de execuções e que envolve um componente interpretativo, o que implica certa subjetividade na interpretação dos resultados. Por essa razão, os achados dessa etapa devem ser entendidos como indicativos e não necessariamente generalizáveis para todos os cenários possíveis.

4 Experimentos e Resultados

Neste capítulo, serão descritos os experimentos realizados para avaliar a eficácia dos LLMs utilizados e do agente desenvolvido ao resolver questões de vestibulares brasileiros. Serão analisados os resultados de testes realizados em diferentes cenários, considerando tanto a acurácia nas respostas quanto a autonomia e eficiência dos agentes em seguir um fluxo lógico ao selecionar e utilizar as ferramentas oferecidas para a realização da tarefa proposta.

4.1 Experimentos iniciais com LLMs

Os primeiros experimentos foram realizados exclusivamente com os LLMs, sem o controle de fluxo ou a adição de ferramentas. Essa etapa inicial é crucial para estabelecer uma base de resultados, permitindo avaliar se um agente baseado em um modelo de linguagem realmente apresenta um desempenho superior na tarefa proposta em comparação ao modelo de linguagem utilizado de forma direta.

Para garantir a consistência das respostas, foi definido um *prompt* específico para conduzir a inferência, além de utilizar o mecanismo de saída estruturada que obriga o modelo a responder apenas com a letra da alternativa correta. Na figura 8, estão especificados tanto o *prompt* quanto a classe que estruturou a saída do modelo.

Figura 8 – *Prompt* e estruturação de resposta para resolução de questão com LLM

Prompt	Você receberá o enunciado de uma pergunta de múltipla escolha e {num_alternatives} alternativas. Dentre as alternativas, APENAS UMA é correta, o restante está errado. Você deve escolher, DENTRE AS ALTERNATIVAS OFERECIDAS, a alternativa que responde corretamente à pergunta. Sua resposta deve ser somente a letra que identifica a alternativa, sem mais comentários ou raciocínio. Enunciado: {question} Alternativas: {alternatives}
Estruturação da Saída	class Answer: option: Literal[“A”, “B”, “C”, “D”, “E”]

Fonte: Elaborada pelo autor

Dessa forma, os LLMs foram submetidos ao teste de resolver as questões sem imagens das 5 provas mais recentes disponíveis da USP (242 questões) e da Unicamp (279 questões), e duas do ENEM (218 questões), o que totaliza 739 questões. O desempenho geral dos modelos

Tabela 1 – Performance dos modelos *Mistral* e *Llama 3.1* na resolução direta de questões dos vestibulares

ENEM			
Modelo	Acertos	Erros	Acurácia (%)
Mistral	122	96	55,96
Llama 3.1	134	84	61,47
USP			
Modelo	Acertos	Erros	Acurácia (%)
Mistral	123	119	50,83
Llama 3.1	139	103	57,44
Unicamp			
Modelo	Acertos	Erros	Acurácia (%)
Mistral	140	139	50,18
Llama 3.1	145	134	51,97

Tabela 2 – Performance dos modelos Mistral e Llama 3.1 por matéria nas questões dos vestibulares USP e Unicamp

USP						
	Mistral			Llama 3.1		
	Total de Perguntas	Acertos	Acurácia (%)	Total de Perguntas	Acertos	Acurácia (%)
Biologia	17	8	47,06	17	11	64,70
Filosofia	10	8	80,00	10	8	80,00
Física	18	5	27,78	18	5	27,78
Geografia	25	13	52,00	25	16	64,00
História	49	31	63,26	49	36	73,46
Inglês	28	22	78,57	28	25	89,28
Matemática	29	6	20,69	29	6	20,69
Português	70	32	45,71	70	34	48,57
Química	15	8	53,33	15	9	60,00
Unicamp						
	Mistral			Llama 3.1		
	Total de Perguntas	Acertos	Acurácia (%)	Total de Perguntas	Acertos	Acurácia (%)
Biologia	29	17	58,62	29	19	65,51
Filosofia	6	5	83,33	6	5	83,33
Física	31	7	22,58	31	9	29,03
Geografia	30	20	66,67	30	19	63,33
História	47	39	82,98	47	40	85,10
Inglês	21	11	52,38	21	10	47,62
Matemática	59	17	28,81	59	16	27,12
Português	60	33	55,00	60	32	53,33
Química	16	6	37,50	16	6	37,50

Fonte: elaborada pelo autor

em cada prova está descrito na tabela 1, e o desempenho por matéria e categoria de questão nas tabelas 2 e 3, respectivamente.

No geral, o modelo *Llama 3.1* se saiu melhor que o Mistral nas três provas. Ambos os modelos se destacaram mais nas questões que exigem entendimento de texto e conhecimento geral de mundo (as categorias TC, EK e TU). Isso mostra que eles são muito bons em ler e usar informações que já possuem.

Tabela 3 – Performance dos modelos Mistral e Llama 3.1 por categoria nas questões dos vestibulares ENEM, USP e Unicamp

	ENEM					
	Mistral			Llama 3.1		
	Total de Perguntas	Acertos	Acurácia (%)	Total de Perguntas	Acertos	Acurácia (%)
@EK	69	46	66,67	69	56	81,16
@TC	166	109	65,66	166	123	74,10
@DS	39	23	58,97	39	26	66,67
@CE	6	3	50,00	6	1	16,67
@MR	45	9	20,00	45	7	15,56
	USP					
	Mistral			Llama 3.1		Llama 3.1
	Total de Perguntas	Acertos	Acurácia (%)	Total de Perguntas	Acertos	Acurácia (%)
BK	49	27	55,10	49	32	65,31
ML	28	22	78,57	28	25	89,28
MR	50	12	24,00	50	12	24,00
PRK	179	84	46,93	179	94	52,51
TU	228	117	51,31	228	130	57,02
	Unicamp					
	Mistral			Llama 3.1		
	Total de Perguntas	Acertos	Acurácia (%)	Total de Perguntas	Acertos	Acurácia (%)
BK	35	23	65,71	35	23	65,71
ML	21	11	52,38	21	10	47,62
MR	95	25	26,31	95	26	27,37
PRK	219	108	49,31	219	114	52,05
TU	239	127	53,14	239	132	55,23

Fonte: elaborada pelo autor

É importante notar que mesmo quando as questões precisavam de um conhecimento prévio, a pontuação deles não caiu muito. Isso sugere que os modelos possuem parte desse conhecimento prévio internalizado, ou conseguem deduzi-lo com base no enunciado das questões. O Llama 3.1 também foi bem nas questões multilíngues da USP – o que faz sentido, já que esses modelos são proficientes em inglês – e teve um resultado satisfatório em perguntas sobre o contexto brasileiro.

O principal problema, ou limitação, para ambos os modelos é o raciocínio matemático, que aparece em todas as provas. No ENEM, a taxa de acerto em MR foi muito baixa (por volta de 20% para o Mistral e 16% para o Llama 3.1). Nas provas da USP e Unicamp, o desempenho em raciocínio matemático aumentou um pouco, mas ainda é baixo (cerca de 24% a 27%). Essa dificuldade se reflete nas matérias de Matemática e Física, onde os resultados foram particularmente ruins.

Além disso, no ENEM, o Llama 3.1 foi inconstante nas questões sobre elementos químicos, com uma queda significativa de acerto nesse grupo. Isso pode indicar que ele é sensível à maneira como as informações sobre química são apresentadas no texto.

Em resumo, a análise mostra que os modelos são ótimos em entender textos e em conhecimento geral, razoáveis quando precisam de conhecimento prévio, mas têm grandes dificuldades em tarefas que exigem cálculos e raciocínio lógico.

4.2 Experimentos com agente

Após o estabelecimento da base de comparação com o uso direto dos LLMs, foram executados os mesmos experimentos com o fluxo do agente coordenado pelo Mistral e Llama 3.1. No apêndice A estão descritos os *prompts* utilizados em cada um dos nós do fluxo do agente durante os experimentos com ambos os modelos.

A tabela 4 demonstra os resultados gerais dos agentes nos experimentos, considerando os fatores de questões não respondidas e redirecionamentos (número de perguntas respondidas pelo nó de contingência do agente). As tabelas 5 e 6 organizam os resultados por matéria e categoria de questão, respectivamente. É importante ressaltar que a acurácia em todas as tabelas se refere a todas as questões, considerando as não respondidas como erros. Isso porque a ideia do agente era, de fato, responder às perguntas e se o nó conclusor falhou em extrair a resposta final, o agente falhou.

Tabela 4 – Performance dos agentes baseados nos modelos Mistral e Llama 3.1 nas questões de vestibular

ENEM						
Modelo	Acertos	Erros	Não respondidas	Acurácia (%)	Quantidade de passos média	Redirecionamentos
Mistral	74	123	21	33,94	6,07	0
Llama 3.1	114	83	21	52,29	15,45	1
USP						
Modelo	Acertos	Erros	Não respondidas	Acurácia (%)	Quantidade de passos média	Redirecionamentos
Mistral	55	167	20	22,73	6,79	0
Llama 3.1	74	150	18	30,58	16,28	0
Unicamp						
Modelo	Acertos	Erros	Não respondidas	Acurácia (%)	Quantidade de passos média	Redirecionamentos
Mistral	70	174	35	25,09	6,74	0
Llama 3.1	84	159	36	30,11	16,21	2

Fonte: elaborada pelo autor.

Tabela 5 – Performance por matéria dos agentes baseados nos modelos Mistral e Llama 3.1 nas questões da USP e Unicamp

USP						
	Mistral			Llama 3.1		
	Total de Perguntas	Acertos	Acurácia (%)	Total de Perguntas	Acertos	Acurácia (%)
Biologia	17	4	23,53	17	3	17,64
Filosofia	10	3	30,00	10	2	20,00
Física	18	6	33,33	18	3	16,67
Geografia	25	7	28,00	25	7	28,00
História	49	11	22,45	49	24	48,98
Inglês	28	12	42,86	28	12	42,85
Matemática	29	6	20,69	29	7	24,13
Português	70	10	14,28	70	18	25,71
Química	15	3	20,00	15	4	26,67
Unicamp						
	Mistral			Llama 3.1		
	Total de Perguntas	Acertos	Acurácia (%)	Total de Perguntas	Acertos	Acurácia (%)
Biologia	29	13	44,83	29	12	41,38
Filosofia	6	3	50,00	6	3	50,00
Física	31	5	16,13	31	5	16,13
Geografia	30	10	33,33	30	11	36,67
História	47	16	34,04	47	17	36,17
Inglês	21	4	19,05	21	6	28,57
Matemática	59	12	20,34	59	6	10,17
Química	16	3	18,75	16	2	12,50
Português	60	25	41,67	60	13	21,67

Fonte: elaborada pelo autor.

Tabela 6 – Performance por categoria dos agentes baseados nos modelos Mistral e Llama 3.1 nas questões de vestibular

ENEM						
	Mistral			Llama 3.1		
	Total de Perguntas	Acertos	Acurácia (%)	Total de Perguntas	Acertos	Acurácia (%)
@TC	166	65	39,16	166	99	59,64
@EK	69	27	39,13	69	39	56,52
@DS	39	12	30,77	39	18	46,15
@CE	6	1	16,67	6	2	33,33
@MR	45	7	15,56	45	11	24,44
USP						
	Mistral			Llama 3.1		
	Total de Perguntas	Acertos	Acurácia (%)	Total de Perguntas	Acertos	Acurácia (%)
ML	28	12	42,86	28	12	42,86
TU	228	51	22,37	228	71	31,14
BK	49	10	20,41	49	16	32,65
PRK	179	35	19,55	179	50	27,93
MR	50	8	16,00	50	12	24,00
Unicamp						
	Mistral			Llama 3.1		
	Total de Perguntas	Acertos	Acurácia (%)	Total de Perguntas	Acertos	Acurácia (%)
BK	35	12	34,28	35	12	34,28
ML	21	6	28,57	21	4	19,05
TU	239	61	25,52	239	77	32,22
PRK	219	60	27,40	219	67	30,59
MR	95	11	11,58	95	16	16,84

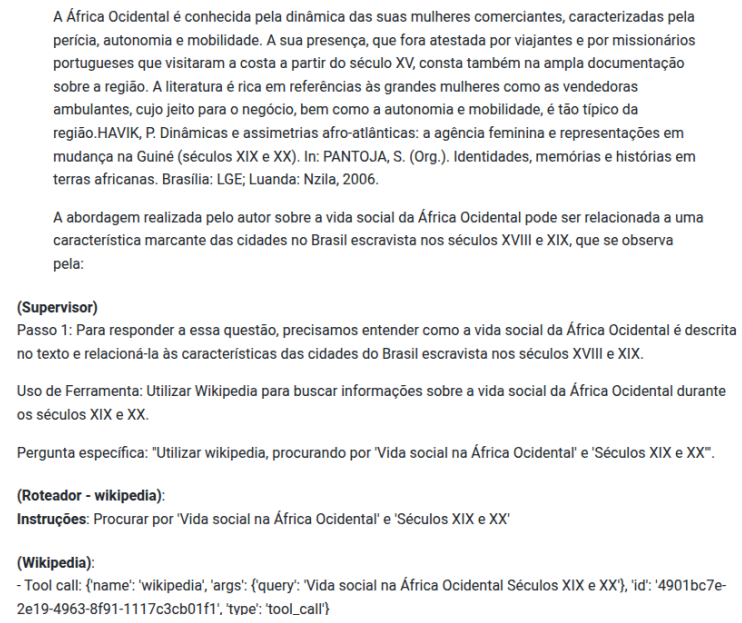
Fonte: elaborada pelo autor.

Em síntese, os agentes baseados em Mistral e Llama 3.1 apresentaram acurácia inferior ao uso direto dos LLMs: no ENEM, reduções de $22{,}02$ e $9{,}18$ pontos percentuais, respectivamente; na USP, de $28{,}10$ (Mistral) e $26{,}86$ (Llama); e na Unicamp, de $25{,}09$ (Mistral) e $21{,}86$ (Llama). As taxas de não resposta, que contam como erro, ficaram entre $7{,}44\%$ e $8{,}26\%$ (USP), em $9{,}63\%$ (ENEM) e entre $12{,}55\%$ e $12{,}90\%$ (Unicamp).

A análise por categoria confirma esse padrão. No ENEM, o agente-Mistral perdeu desempenho em todas as categorias, com queda menor em MR. O agente-Llama também caiu em TC, EK e DS, mas apresentou ganhos em CE e MR. Por matéria, vale destacar que, em Matemática, na USP, o agente-Llama apresentou ligeira melhora, enquanto Inglês e História/USP teve grande piora. Já na Unicamp com o Llama, Português e Matemática tiveram quedas significativas. Esses resultados indicam que, na configuração testada, a estrutura em múltiplos passos introduziu novos pontos de falha que, na média, não compensaram os potenciais ganhos de organização do raciocínio — com exceções pontuais, como em MR e CE no ENEM com Llama.

Entretanto, o comportamento do agente, principalmente em relação ao uso de ferramentas e organização de raciocínio se apresentou satisfatório. Por exemplo, na figura 9 é possível observar que, a partir do enunciado da questão, o modelo começou seu raciocínio por uma busca pelo termo relacionado "vida social na áfrica", evidenciando capacidade de formular consultas úteis e de encadear a leitura de evidências textuais antes de decidir. O fluxo completo da resolução dessa questão está descrito no apêndice B.

Figura 9 – Execução do agente com uso da Wikipédia

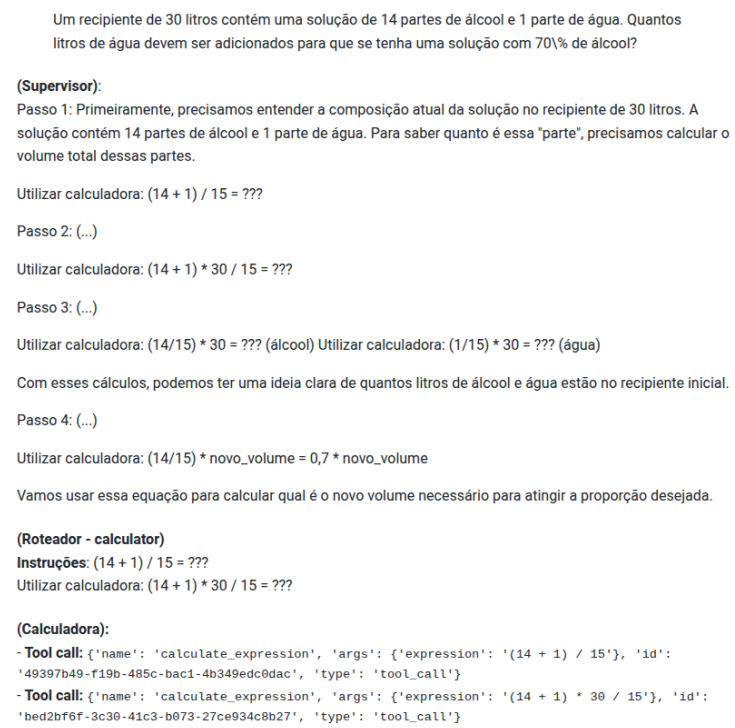


Fonte: Elaborada pelo autor

Outro exemplo de raciocínio estruturado está ilustrado na figura 10. Nessa questão,

o agente falhou em seguir as instruções de executar um passo por vez, mas construiu um raciocínio organizado em alguns passos que eram definidos por algumas equações. Foram passadas três equações para serem resolvidas, porém o nó roteador transferiu instruções para somente duas delas, expondo uma fragilidade no controle do fluxo. A resolução completa dessa questão pelo agente está no apêndice C.

Figura 10 – Execução do agente com uso da Calculadora



Fonte: Elaborada pelo autor

Em síntese, os exemplos sugerem que o desenho do agente contribui para a organização do processo, mesmo que a resposta final não seja alcançada por determinados motivos.

4.3 Discussão dos resultados

Os resultados indicam que a configuração de agente, na configuração testada, não converteu o potencial de estruturação do raciocínio em ganhos efetivos de acurácia. Duas hipóteses centrais ajudam a explicar esse problema.

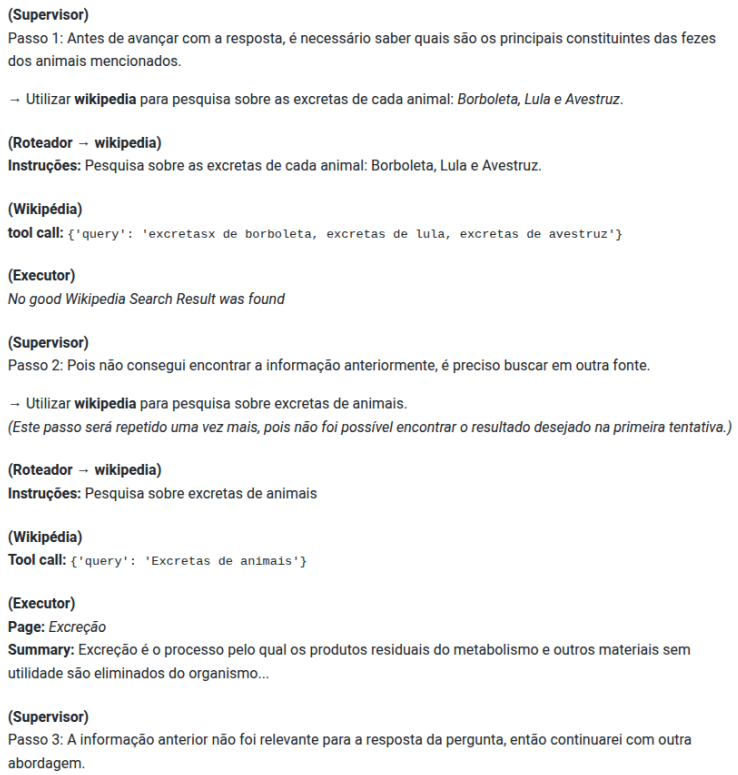
4.3.1 Limitações da ferramenta de Wikipédia

A ferramenta de Wikipédia mostrou-se pouco adequada para o domínio exigidos por questões de vestibular. Em primeiro lugar, houve dificuldade na busca: os termos de busca definidos pelo supervisor muitas vezes acabaram por não gerar resultado algum, ou por gerar

resultados de assuntos totalmente diferentes do esperado. Isso faz com que o agente fique preso no mesmo passo de resolução por muito tempo e pode até causar confusão durante a resolução, se informações não relacionadas forem trazidas e tomadas como verdadeiras pelo agente.

Em segundo lugar, observou-se pouca especificidade do conteúdo recuperado: quando havia retorno, o texto não trazia valores, definições ou relações realmente úteis, o que forçou o agente a inferências frágeis. O caso da questão da USP sobre excretas, demonstrado pela figura 11 (fluxo completo registrado no apêndice E), ilustra ambos os problemas da ferramenta, a busca pelos termos específicos determinados pelo modelo não retornou conteúdo, e a busca pelo termo mais genérico retornou informações que o modelo considerou inúteis para o contexto da questão.

Figura 11 – Exemplo de execução do agente no qual a ferramenta da Wikipédia falhou



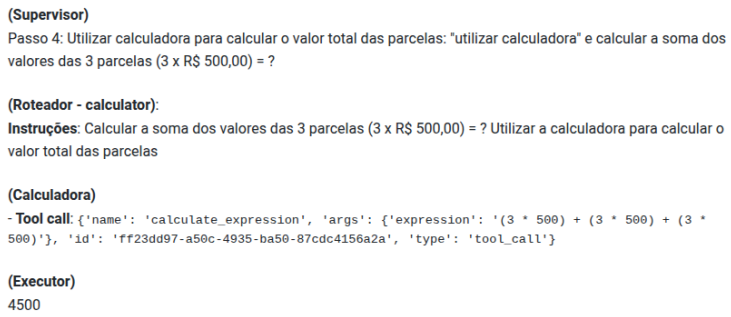
Fonte: Elaborada pelo autor

4.3.2 Tamanho dos modelos e controle do fluxo

O tamanho (e, por consequência, a capacidade) dos modelos impactou o fluxo, principalmente nas etapas em que o LLM precisa interpretar saídas geradas por outro LLM, como nos nós especializados de ferramentas, o roteador e o conclusor. Um exemplo desse problema é demonstrado na figura 12 (fluxo completo registrado no apêndice D): o nó especializado

na ferramenta de calculadora interpretou o cálculo solicitado pelo supervisor de forma errada, retornando um valor diferente e causando impacto em todo o restante do processo de resolução.

Figura 12 – Parte da execução do agente com falha de interpretação



Fonte: Elaborada pelo autor

Para explorar essa hipótese, os experimentos foram repetidos com o modelo GPT 4.1 Mini, da OpenAI (OpenAI, 2025), que é maior em questão de parâmetros e também se diferencia por possuir uma janela de contexto de mais de um milhão de *tokens*, o que facilita a interpretação até de textos mais longos e também evita a perda de informação entre as várias etapas do fluxo. Esse modelo da OpenAI não era uma das primeiras opções para o teste do fluxo de agente por ser de código fechado e, consequentemente, ter seu uso pago. Entretanto, ele se apresentou como uma boa alternativa para gerar uma base de comparação e analisar se o problema realmente está no fluxo desenhado ou nos modelos utilizados para teste. Os resultados dos experimentos realizados com o GPT 4.1 Mini estão na tabela 7.

Em acurácia geral, o agente com GPT-4.1 Mini ficou abaixo do uso direto do próprio modelo. Ainda assim, quando a tarefa exigiu cálculo, o agente com GPT trouxe ganhos expressivos, chegando a $88{,}13\%$ em Matemática e a 100% em Física na Unicamp, evidenciando o melhor aproveitamento da calculadora. As perdas concentraram-se nas áreas textuais, como em Filosofia/USP e Português/USP.

Comparando agentes, o baseado no GPT superou amplamente seus pares com Llama e Mistral em todas as bancas, como descrito na tabela 8. A conclusão imediata é que o baixo desempenho observado com Mistral e Llama decorre mais de limitações dos modelos e da recuperação de evidências do que de problemas estruturais no fluxo. Quando pareado a um modelo de maior capacidade, o agente mostra-se competitivo na média e claramente superior em tarefas quantitativas, o que evidencia ainda mais a lacuna deixada pela ferramenta da Wikipédia.

Tabela 7 – Performance do modelo GPT 4.1 Mini

	Modelo		Agente				
	Acertos	Acurácia (%)	Acertos	Erros	Sem resposta	Acurácia (%)	Média de passos
ENEM	180	82,57	148	21	49	67,89	14,26
USP	188	77,68	159	27	56	65,70	15,15
Unicamp	201	72,04	185	39	55	66,30	16,28
USP - Matérias							
	Total de Questões	Acertos	Acurácia (%)	Total de Questões	Acertos	Erros	Acurácia (%)
Biologia	17	14	82,35	17	11	3	64,70
Filosofia	10	10	100	10	4	2	40,00
Física	18	14	77,78	18	17	1	94,44
Geografia	25	21	84,00	25	16	5	64,00
História	49	45	91,84	49	32	5	65,31
Inglês	28	27	96,42	28	20	3	71,43
Matemática	29	8	27,59	29	24	2	82,76
Português	70	55	78,57	70	34	7	48,57
Química	15	12	80,00	15	12	0	80,00
Unicamp - Matérias							
	Total de Questões	Acertos	Acurácia (%)	Total de Questões	Acertos	Erros	Acurácia (%)
Biologia	29	27	93,10	29	22	3	75,86
Filosofia	6	6	100	6	3	0	50,00
Física	31	16	51,61	31	31	0	100
Geografia	30	27	90,00	30	20	3	66,67
História	47	46	97,87	47	30	4	63,83
Inglês	21	19	90,47	21	11	4	52,38
Matemática	59	24	40,68	59	52	3	88,13
Português	60	46	76,67	60	21	18	35,00
Química	16	9	56,25	16	8	6	50,00

Fonte: elaborada pelo autor.

Tabela 8 – Comparação da performance do GPT 4.1 Mini com Mistral e Llama 3.1

Modelo	Acurácia (%)		
	Enem	USP	Unicamp
Mistral	55,96	50,83	50,18
Llama 3.1	61,47	57,44	51,97
GPT 4.1 Mini	82,57	77,68	72,04
Agente	Acurácia (%)		
	Enem	USP	Unicamp
Mistral	33,94	22,73	25,09
Llama 3.1	52,29	30,58	30,11
GPT 4.1 Mini	67,89	65,70	66,30

Fonte: elaborada pelo autor.

De forma geral, os resultados indicam que a simples introdução de um fluxo estruturado não garante ganhos automáticos de desempenho em relação ao uso direto dos modelos, sobretudo quando há limitações na recuperação de evidências ou na capacidade interpretativa do LLM. Ainda assim, os experimentos revelam potencial significativo do agente em cenários que exigem raciocínio procedural e uso de ferramentas, especialmente quando combinado a modelos mais robustos

5 Conclusão

O presente trabalho teve como objetivo investigar o uso de agentes inteligentes baseados em LLMs para a resolução de questões de vestibulares brasileiros. O domínio selecionado apresenta alta complexidade, por envolver habilidades que combinam interpretação de textos extensos, conhecimento multidisciplinar, raciocínio matemático e compreensão contextual, que são os tipos de tarefas em que os LLMs ainda demonstram limitações relevantes (ZHANG et al., 2023) (AHN et al., 2024).

Nesse contexto, foi proposta uma arquitetura de agente que realiza a decomposição da tarefa em etapas sucessivas e integra o LLM a ferramentas externas específicas, sendo uma calculadora e uma enciclopédia. A avaliação foi conduzida a partir de conjuntos de dados compostos por provas do ENEM (SILVEIRA; MAUÁ, 2017), USP e Unicamp (ALMEIDA et al., 2023), comparando-se o desempenho do agente com o uso direto dos modelos nas mesmas questões. Os experimentos foram realizados com os modelos Llama 3.1 e Mistral, ambos gratuitos e relativamente pequenos, para uso local.

Os resultados apontaram que o agente coordenado por esses modelos não alcançou bom desempenho, apresentando acurácia consistentemente inferior ao uso direto dos LLMs em todas as áreas do conhecimento e bancas avaliadas. Ainda assim, observou-se que a estrutura do fluxo exibia comportamento promissor, pois em diversos exemplos, o agente demonstrou capacidade de formular um raciocínio coerente e começar a executá-lo com o uso das ferramentas. Diante disso, foram levantadas duas hipóteses principais para o desempenho abaixo do esperado: i) a ferramenta de enciclopédia frequentemente não retornava informações úteis ao contexto da questão; e ii) o agente falhava, sobretudo, nas etapas que exigiam interpretar conteúdos produzidos pelo próprio LLM em etapas anteriores. Para investigar essas limitações, os experimentos foram repetidos utilizando um modelo substancialmente mais robusto: o GPT-4.1 Mini.

Os resultados com o GPT-4.1 Mini reforçaram as hipóteses levantadas. Embora o agente ainda ficasse ligeiramente abaixo do uso direto do próprio modelo em termos de acurácia média, ele apresentou ganhos expressivos nas tarefas que dependiam de raciocínio quantitativo. Esse comportamento indica que, quando a ferramenta utilizada é suficientemente competente para resolver o problema proposto, a arquitetura pode, potencializar a resolução. Em contrapartida, observou-se queda relevante em tarefas predominantemente textuais, sugerindo que a organização do fluxo pode limitar a capacidade natural dos LLMs de interpretação de textos.

A principal contribuição deste trabalho consiste, portanto, na validação experimental de que a organização de um agente baseado em LLM não garante, por si só, ganhos de

desempenho sobre o seu uso direto, especialmente quando se acrescentam fatores limitantes como tamanho do modelo e ferramentas pouco específicas para o contexto. No entanto, os resultados mostram que a abordagem tem potencial significativo quando combinada a modelos mais robustos e a ferramentas especializadas para o domínio, sobretudo em tarefas que exigem raciocínio matemático.

Como próximos passos, propõe-se a melhoria das ferramentas externas, principalmente quando se trata de consultas factuais, explorando abordagens de RAG ou outros serviços que ofereçam buscas mais específicas ou informações mais diretas. Além disso, também é levantada a possibilidade de explorar o potencial de modelos pequenos como coordenadores de agentes inteligentes de outra forma, com sumarização de informações e a diminuição de etapas interpretativas nos fluxos propostos. Esses aprimoramentos podem tornar o agente mais preciso, mais confiável e próximo de aplicações reais em contextos educacionais.

Referências

AHN, J.; VERMA, R.; LOU, R.; LIU, D.; ZHANG, R.; YIN, W. *Large Language Models for Mathematical Reasoning: Progresses and Challenges*. 2024. Disponível em: <https://arxiv.org/abs/2402.00157>. Acesso em: 16 nov. 2025.

ALMEIDA, T. S.; LAITZ, T.; BONáS, G. K.; NOGUEIRA, R. *BLUEX: A benchmark based on Brazilian Leading Universities Entrance eXams*. 2023.

BROWN, T. B.; MANN, B.; RYDER, N.; SUBBIAH, M.; KAPLAN, J.; DHARIWAL, P.; NEELAKANTAN, A.; SHYAM, P.; SASTRY, G.; ASKELL, A.; AGARWAL, S.; HERBERT-VOSS, A.; KRUEGER, G.; HENIGHAN, T.; CHILD, R.; RAMESH, A.; ZIEGLER, D. M.; WU, J.; WINTER, C.; HESSE, C.; CHEN, M.; SIGLER, E.; LITWIN, M.; GRAY, S.; CHESS, B.; CLARK, J.; BERNER, C.; MCCANDLISH, S.; RADFORD, A.; SUTSKEVER, I.; AMODEI, D. *Language Models are Few-Shot Learners*. 2020. Disponível em: <https://arxiv.org/abs/2005.14165>. Acesso em: 16 nov. 2025.

CHU, Z.; WANG, S.; XIE, J.; ZHU, T.; YAN, Y.; YE, J.; ZHONG, A.; HU, X.; LIANG, J.; YU, P. S.; WEN, Q. *LLM Agents for Education: Advances and Applications*. 2025. Disponível em: <https://arxiv.org/abs/2503.11733>. Acesso em: 16 nov. 2025.

GAO, Y.; XIONG, Y.; GAO, X.; JIA, K.; PAN, J.; BI, Y.; DAI, Y.; SUN, J.; WANG, M.; WANG, H. *Retrieval-Augmented Generation for Large Language Models: A Survey*. 2024. Disponível em: <https://arxiv.org/abs/2312.10997>. Acesso em: 16 nov. 2025.

GRATTAFIORI, A. et al. *The Llama 3 Herd of Models*. 2024. Disponível em: <https://arxiv.org/abs/2407.21783>. Acesso em: 16 nov. 2025.

JIANG, A. Q.; SABLAYROLLES, A.; MENSCH, A.; BAMFORD, C.; CHAPLOT, D. S.; CASAS, D. de las; BRESSAND, F.; LENGYEL, G.; LAMPLE, G.; SAULNIER, L.; LAVAUD, L. R.; LACHAUX, M.-A.; STOCK, P.; SCAO, T. L.; LAVRIL, T.; WANG, T.; LACROIX, T.; SAYED, W. E. *Mistral 7B*. 2023. Disponível em: <https://arxiv.org/abs/2310.06825>. Acesso em: 16 nov. 2025.

JOVANOVIC, M.; CAMPBELL, M. Compacting ai: In search of the small language model. *Computer*, v. 57, 08 2024.

LEWIS, P.; PEREZ, E.; PIKTUS, A.; PETRONI, F.; KARPUKHIN, V.; GOYAL, N.; KüTTLER, H.; LEWIS, M.; YIH, W. tau; ROCKTäSCHEL, T.; RIEDEL, S.; KIELA, D. *Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks*. 2021. Disponível em: <https://arxiv.org/abs/2005.11401>. Acesso em: 16 nov. 2025.

NUNES, D.; PRIMI, R.; PIRES, R.; LOTUFO, R.; NOGUEIRA, R. *Evaluating GPT-3.5 and GPT-4 Models on Brazilian University Admission Exams*. 2023. Disponível em: <https://arxiv.org/abs/2303.17003>. Acesso em: 16 nov. 2025.

OpenAI. *Introducing GPT-4.1 in the API*. 2025. Accessed: 2025-10-27. Disponível em: <https://openai.com/index/gpt-4-1/>. Acesso em: 16 nov. 2025.

PANG, X.; HONG, R.; ZHOU, Z.; LV, F.; YANG, X.; LIANG, Z.; HAN, B.; ZHANG, C. *Physics Reasoner: Knowledge-Augmented Reasoning for Solving Physics Problems with Large Language Models*. 2024. Disponível em: <https://arxiv.org/abs/2412.13791>. Acesso em: 16 nov. 2025.

RUSSELL, S.; NORVIG, P. *Artificial Intelligence: A Modern Approach*. 3rd. ed. USA: Prentice Hall Press, 2009. ISBN 0136042597.

SHEN, Z. *LLM With Tools: A Survey*. 2024. Disponível em: <https://arxiv.org/abs/2409.18807>. Acesso em: 16 nov. 2025.

SILVEIRA, I. C.; MAUÁ, D. D. University entrance exam as a guiding test for artificial intelligence. In: *Proceedings of the 6th Brazilian Conference on Intelligent Systems*. [S.l.: s.n.], 2017. (BRACIS), p. 426–431.

SUPERBI, J.; PINTO, H.; SANTOS, E.; LATTARI, L.; CASTRO, B. Enhancing large language model performance on enem math questions using retrieval-augmented generation. In: *Anais do XVIII Brazilian e-Science Workshop*. Porto Alegre, RS, Brasil: SBC, 2024. p. 56–63. ISSN 2763-8774. Disponível em: <https://sol.sbc.org.br/index.php/bresci/article/view/30581>. Acesso em: 16 nov. 2025.

VASWANI, A.; SHAZEER, N.; PARMAR, N.; USZKOREIT, J.; JONES, L.; GOMEZ, A. N.; KAISER, L.; POLOSUKHIN, I. *Attention Is All You Need*. 2023. Disponível em: <https://arxiv.org/abs/1706.03762>. Acesso em: 16 nov. 2025.

WEI, J.; TAY, Y.; BOMMASANI, R.; RAFFEL, C.; ZOPH, B.; BORGEAUD, S.; YOGATAMA, D.; BOSMA, M.; ZHOU, D.; METZLER, D.; CHI, E. H.; HASHIMOTO, T.; VINYALS, O.; LIANG, P.; DEAN, J.; FEDUS, W. *Emergent Abilities of Large Language Models*. 2022. Disponível em: <https://arxiv.org/abs/2206.07682>. Acesso em: 16 nov. 2025.

WEI, J.; WANG, X.; SCHUURMANS, D.; BOSMA, M.; ICHTER, B.; XIA, F.; CHI, E.; LE, Q.; ZHOU, D. *Chain-of-Thought Prompting Elicits Reasoning in Large Language Models*. 2023. Disponível em: <https://arxiv.org/abs/2201.11903>. Acesso em: 16 nov. 2025.

WOOLDRIDGE, M.; JENNINGS, N. R. Intelligent agents: theory and practice. *The Knowledge Engineering Review*, v. 10, n. 2, p. 115–152, 1995.

WU, Y.; JIA, F.; ZHANG, S.; LI, H.; ZHU, E.; WANG, Y.; LEE, Y. T.; PENG, R.; WU, Q.; WANG, C. *MathChat: Converse to Tackle Challenging Math Problems with LLM Agents*. 2024. Disponível em: <https://arxiv.org/abs/2306.01337>. Acesso em: 16 nov. 2025.

XI, Z.; CHEN, W.; GUO, X.; HE, W.; DING, Y.; HONG, B.; ZHANG, M.; WANG, J.; JIN, S.; ZHOU, E.; ZHENG, R.; FAN, X.; WANG, X.; XIONG, L.; ZHOU, Y.; WANG, W.; JIANG, C.; ZOU, Y.; LIU, X.; YIN, Z.; DOU, S.; WENG, R.; CHENG, W.; ZHANG, Q.; QIN, W.; ZHENG, Y.; QIU, X.; HUANG, X.; GUI, T. *The Rise and Potential of Large Language Model Based Agents: A Survey*. 2023. Disponível em: <https://arxiv.org/abs/2309.07864>. Acesso em: 16 nov. 2025.

ZHANG, Y.; LI, Y.; CUI, L.; CAI, D.; LIU, L.; FU, T.; HUANG, X.; ZHAO, E.; ZHANG, Y.; XU, C.; CHEN, Y.; WANG, L.; LUU, A. T.; BI, W.; SHI, F.; SHI, S. *Siren's Song in the AI Ocean: A Survey on Hallucination in Large Language Models*. 2023. Disponível em: <https://arxiv.org/abs/2309.01219>. Acesso em: 16 nov. 2025.

ÖRPEK, Z.; TURAL, B.; DESTAN, Z. The language model revolution: Llm and slm analysis. In: *2024 8th International Artificial Intelligence and Data Processing Symposium (IDAP)*. [S.l.: s.n.], 2024. p. 1–4.

Apêndices

APÊNDICE A – Prompts e saídas estruturadas para o fluxo do agente

Supervisor — *prompt* e estrutura de saída

Prompt	*Estrutura da saída*
Você é um Agente Coordenador encarregado de resolver questões de múltipla escolha utilizando exclusivamente duas ferramentas externas: calculator e wikipedia. IMPORTANTE**:** Você não executa as ferramentas e nunca deve presumir ou inventar os resultados de cálculos ou buscas. Seu trabalho é estruturar o problema e solicitar a informação necessária. 1. Fluxo de Resolução (Passo a Passo) ▪ Responda com um único passo de raciocínio por vez. ▪ Não execute mais de um passo de uma só vez; avance passo a passo. ▪ Retorne apenas um passo e o uso da ferramenta correspondente. ▪ Utilize as ferramentas para consultar informações e fazer cálculos. ▪ Não retorne a resposta final até concluir os passos e o uso necessário das ferramentas. 2. Uso de Ferramentas ▪ Calculadora: escreva "utilizar calculadora" e especifique o cálculo. ▪ Wikipédia: escreva "utilizar wikipedia" e especifique o tópico. 3. Gestão de Erros e Finalização ▪ Resultado vazio: tente novamente no próximo passo, refinando a busca ou o cálculo. ▪ Resposta final: ao concluir, escreva "Resposta Final:" seguido da alternativa (letra).	Texto plano

Elaborado pelo autor.

Roteador — *prompt* e estrutura de saída

Prompt	*Estrutura da saída*
Você é responsável por interpretar a resposta do modelo de linguagem e definir o próximo passo. O modelo está resolvendo uma questão de múltipla escolha e pode: ▪ pedir para usar a calculadora; ▪ pedir para usar a wikipédia (ferramenta de busca); ▪ retornar "RESPOSTA FINAL". Com base nisso, defina: ▪ route: calculator, wikipedia ou END; ▪ route_instructions: instruções a serem passadas diretamente à ferramenta. Regras ▪ A rota só será END se houver "RESPOSTA FINAL" explícita. ▪ Caso contrário, encaminhe para uma ferramenta. ▪ Não interfira no conteúdo; apenas repasse as instruções.	Objeto com propriedades: ▪ `route` (calculator, wikipedia ou END) ▪ `route_instructions` (texto plano)

Elaborado pelo autor.

Conclusor — *prompt* e estrutura de saída

Prompt	*Estrutura da saída*
Você determinará qual alternativa foi escolhida como correta em uma questão de múltipla escolha. Com base na mensagem recebida, identifique a alternativa. Caso não seja possível identificar com clareza, retorne "-".	Uma letra de A a E, ou "-"

Elaborado pelo autor.

APÊNDICE B – Exemplo de execução do agente: ENEM 2016, questão 15

Questão e alternativas

A África Ocidental é conhecida pela dinâmica das suas mulheres comerciantes, caracterizadas pela perícia, autonomia e mobilidade. A sua presença, que fora atestada por viajantes e por missionários portugueses que visitaram a costa a partir do século XV, consta também na ampla documentação sobre a região. A literatura é rica em referências às grandes mulheres como as vendedoras ambulantes, cujo jeito para o negócio, bem como a autonomia e mobilidade, é tão típico da região.
A abordagem realizada pelo autor sobre a vida social da África Ocidental pode ser relacionada a uma característica marcante das cidades no Brasil escravista nos séculos XVIII e XIX, que se observa pela:
A: restrição à realização do comércio ambulante por africanos escravizados e seus descendentes. B: convivência entre homens e mulheres livres, de diversas origens, no pequeno comércio. C: presença de mulheres negras no comércio de rua de diversos produtos e alimentos. D: dissolução dos hábitos culturais trazidos do continente de origem dos escravizados. E: entrada de imigrantes portugueses nas atividades ligadas ao pequeno comércio urbano

Supervisor

Passo 1: Para responder a essa questão, precisamos entender como a vida social da África Ocidental é descrita no texto e relacioná-la às características das cidades do Brasil escravista nos séculos XVIII e XIX.
Uso de Ferramenta: Utilizar Wikipedia para buscar informações sobre a vida social da África Ocidental durante os séculos XIX e XX.
Pergunta específica: "Utilizar wikipedia, procurando por 'Vida social na África Ocidental' e 'Séculos XIX e XX'".

Roteador

Roteando para <Wikipedia> com instruções: Procurar por 'Vida social na África Ocidental' e 'Séculos XIX e XX'

Wikipedia

```
Tool call: {name: 'wikipedia', args: {query: 'Vida social na África
Ocidental Séculos XIX e XX'}}
```

Executor

Page: Século XIX Summary: O século XIX começou no dia 1 de janeiro de 1801 e terminou no dia 31 de dezembro de 1900. Foi um período histórico marcado pelo fim dos impérios [...texto truncado...]

Supervisor

Passo 2: Agora que conhecemos um pouco mais sobre a vida social da África Ocidental durante os séculos XIX e XX, precisamos relacioná-la às características das cidades do Brasil escravista nos mesmos períodos.
Uso de Ferramenta: Utilizar Wikipedia para buscar informações sobre as cidades do Brasil escravista nos séculos XVIII e XIX.
Pergunta específica: "Utilizar wikipedia, procurando por 'Cidades do Brasil escravista' e 'Séculos XVIII e XIX'".

Roteador

Roteando para <Wikipedia> com instruções: Buscar por "Cidades do Brasil escravista" e "Séculos XVIII e XIX" na Wikipedia.

Wikipedia

```
Tool call: {name: 'wikipedia', args: {query: 'Cidades do Brasil
escravista Séculos XVIII e XIX'}}
```

Executor

Page: Escravidão no Brasil Summary: A escravidão no Brasil [...texto truncado...]

Supervisor

Passo 3: Agora que temos informações sobre as cidades do Brasil escravista nos séculos XVIII e XIX, podemos relacioná-las à abordagem realizada pelo autor sobre a vida social da África Ocidental.
Uso de Ferramenta: Utilizar Wikipedia para buscar informações sobre o comércio ambulante na África Ocidental.
Pergunta específica: "Utilizar wikipedia, procurando por 'Comércio ambulante na África Ocidental'".

Roteador

Roteando para <Wikipedia> com instruções: Utilizar wikipedia, procurando por 'Comércio ambulante na África Ocidental'.

Wikipedia

`Tool call: {name: 'wikipedia', args: {query: 'Comércio ambulante na África Ocidental'}}`

Executor

Page: Quitandeira Summary: Quitandeira, também chamada de zungueira, é uma vendedora ambulante típica de Angola [...texto truncado...]

Supervisor

Resposta Final: A resposta correta é a alternativa C: presença de mulheres negras no comércio de rua de diversos produtos e alimentos.

Roteador

Roteando para <END> com instruções: A resposta correta é a alternativa C: presença de mulheres negras no comércio de rua de diversos produtos e alimentos.

Conclusor

C

APÊNDICE C – Exemplo de execução do agente: Unicamp 2023, questão 61

Questão e alternativas

Um recipiente de 30 litros contém uma solução de 14 partes de álcool e 1 parte de água. Quantos litros de água devem ser adicionados para que se tenha uma solução com 70% de álcool?
a) 8 litros. b) 10 litros. c) 12 litros. d) 14 litros.

Supervisor

Passo 1: Primeiramente, precisamos entender a composição atual da solução no recipiente de 30 litros. A solução contém 14 partes de álcool e 1 parte de água. Para saber quanto é essa "parte", precisamos calcular o volume total dessas partes.
Utilizar calculadora: (14 + 1) / 15 = ???
Essa expressão nos dá a fração do recipiente que representa cada uma das partes da solução atual.
Passo 2: Em seguida, precisamos saber quanto é cada parte na solução atual para entender melhor como está composta a solução e facilitar os cálculos subsequentes. Entendo que o volume de cada parte é igualmente representado, então podemos usar o resultado do passo anterior para calcular.
Utilizar calculadora: (14 + 1) * 30 / 15 = ???
Esse cálculo nos dá a quantidade em litros que representa 14 partes da solução no recipiente. Agora sabemos que cada parte é igualmente representada por esse volume.
Passo 3: Agora precisamos calcular quanto álcool e água há na solução atual para entendermos a composição correta, pois a pergunta fala em uma solução com "70
Utilizar calculadora: (14/15) * 30 = ??? (álcool) Utilizar calculadora: (1/15) * 30 = ??? (água)
Passo 4: A pergunta fala em adicionar água para que a solução tenha 70
Utilizar calculadora: (14/15) * novo$_v olume = 0,7 * novo_v olume$

Roteador

Roteando para `<Calculator>` com instruções: (14 + 1) / 15 = ??? Utilizar calculadora: (14 + 1) * 30 / 15 = ???

Calculator

`Tool call: {expression: '(14 + 1) / 15'} Tool call: {expression: '(14 + 1) * 30 / 15'}`

Executor

1 30

Supervisor

Essa expressão nos dá o que queremos: o fator de escala aplicado às partes da solução, agora considerando o recipiente como um todo.
Passo 5: Agora precisamos calcular qual é o volume atual do álcool na solução.
Utilizar calculadora: (14/15) * 30 = ???
Passo 6: Agora precisamos calcular quanto álcool e água há na solução atual.
Utilizar calculadora: (14/15) * 30 = ??? (álcool) Utilizar calculadora: (1/15) * 30 = ??? (água)
Passo 7: Quantos litros devem ser adicionados para que 70
Passo 8: Calcular quantos litros de água devem ser adicionados:
Utilizar calculadora: água$_nova = (1/15) * novo_volume - (1/15) * 30 =$???

Roteador

Roteando para <Calculator> com instruções: Calcular a quantidade em litros que representa 14 partes da solução no recipiente: (14/15) * 30 = ???

Calculator

`Tool call: {expression: '(14/15) * 30'}`

Executor

28

Supervisor

Resposta Final: d) 14 litros.
Lembre-se, apenas uma das alternativas é a correta.

Roteador

Roteando para <END> com instruções: Nenhuma ação necessária. A resposta final foi fornecida pelo modelo de linguagem e pode ser considerada concluída nesse estágio.

Conclusor

D

APÊNDICE D – Exemplo de execução do agente: USP 2018, questão 31

Pergunta e alternativas

Maria quer comprar uma TV que está sendo vendida por R$ 1.500,00 à vista ou em 3 parcelas mensais sem juros de R$ 500,00. O dinheiro que Maria reservou para essa compra não é suficiente para pagar à vista, mas descobriu que o banco oferece uma aplicação financeira que rende 1% ao mês. Após fazer os cálculos, Maria concluiu que, se pagar a primeira parcela e, no mesmo dia, aplicar a quantia restante, conseguirá pagar as duas parcelas que faltam sem ter que colocar nem tirar um centavo sequer. Quanto Maria reservou para essa compra, em reais
a) 1.450,20, b) 1.480,20, c) 1.485,20, d) 1.495,20, e) 1.490,20

Supervisor

Passo 1: Para resolver este problema, precisamos entender o cenário de Maria e como ela consegue pagar as parcelas sem ter que colocar ou tirar dinheiro. Primeiramente, vamos considerar a quantia total que ela pagará se optar pela opção de parcelamento. Ela paga R$ 1.500,00 à vista ou em três parcelas de R$ 500,00 cada.
Passo 2: Utilizar calculadora para calcular o valor total das parcelas: "utilizar calculadora"e calcular a soma dos valores das 3 parcelas (3 x R$ 500,00).
Vamos prosseguir com os passos subsequentes.

Roteador

Roteando para `<calculator>` com instruções: Utilizar calculadora para calcular o valor total das parcelas: "utilizar calculadora"e calcular a soma dos valores das 3 parcelas (3 x R$ 500,00).
Vamos prosseguir com os passos subsequentes.

Calculadora

```
Tool call: {name: 'calculate_expression', args: {expression: '3 x R$ 500,00'}}
```

Executor

"Wolfram|Alpha did not understand your input"

Supervisor

Passo 3: Ainda que não tenhamos conseguido realizar o cálculo anteriormente, vamos tentar novamente e reformular a busca para obter o resultado desejado.
Passo 4: Utilizar calculadora para calcular o valor total das parcelas: "utilizar calculadora"e calcular a soma dos valores das 3 parcelas (3 x R$ 500,00) = ?
Vamos prosseguir com os passos subsequentes.

Roteador

Roteando para <calculator> com instruções: Calcular a soma dos valores das 3 parcelas (3 x R$ 500,00) = ? Utilizar a calculadora para calcular o valor total das parcelas.

Calculadora

```
Tool call: {name: 'calculate_expression', args: {expression: '(3 *
500) + (3 * 500) + (3 * 500)'}}
```

Executor

4500

Supervisor

Passo 5: Com o valor total das parcelas conhecido, Maria decide pagar a primeira parcela de R$ 500,00 e aplicar o restante no banco para obter um rendimento de 1% ao mês.
Passo 6: Utilizar Wikipedia para buscar informações sobre juros compostos ou equivalente: "utilizar wikipedia"e procurar por "juros compostos".
Vamos prosseguir com os passos subsequentes.

Roteador

Roteando para <wikipedia> com instruções: Utilizar Wikipedia para buscar informações sobre juros compostos ou equivalente: "utilizar wikipedia"e procurar por "juros compostos".

Wikipedia

```
Tool call: {name: 'wikipedia', args: {query: 'juros compostos'}}
```

Executor

Page: Juro Summary: Juro é a remuneração cobrada pelo empréstimo de dinheiro (ou outro item). É expresso como um percentual sobre o valor emprestado (taxa de juro).

Supervisor

Passo 7: Com a aplicação financeira renderendo 1% ao mês, podemos considerar que o valor da segunda parcela é R$ 4500,00 e vamos calcular como o rendimento dessa aplicação irá cobrir o valor necessário para pagar as duas parcelas restantes.
Passo 8: Utilizar calculadora para calcular a quantidade de dinheiro necessário após um mês com juros compostos: "utilizar calculadora"e calcular (1 + 0,01)^12 * 4500.
Vamos prosseguir com os passos subsequentes.

Roteador

Roteando para <calculator> com instruções: calcular (1 + 0,01)^12 * 4500

Calculadora

```
Tool call: {name: 'calculate_expression', args: {expression: '(1 +
0.01)^12 * 4500'}}
```

Executor

5070.71

Supervisor

Passo 9: Com o valor da segunda parcela após um mês de aplicação financeira conhecido, podemos calcular a quantidade total que Maria reservou para essa compra.
Passo 10: Utilizar calculadora para calcular a diferença entre o valor necessário e o valor que ela já paga: "utilizar calculadora"e calcular 4500 + (1+0.01)^12 * 4500 - 1500.
Vamos prosseguir com os passos subsequentes, realizando a última etapa para obtermos a resposta final.

Roteador

Roteando para <calculator> com instruções: calcular a diferença entre o valor necessário e o valor que ela já paga: "4500 + (1+0.01)^12 * 4500 - 1500"

Calculadora

```
Tool call: {name: 'calculate_expression', args: {expression: '"4500 +
(1+0.01)^12 * 4500 - 1500"'}}
```

Executor

8070.71

Supervisor

Resposta Final: Não consegui determinar o valor que Maria reservou para essa compra, pois, em um dos passos acima, o resultado foi 5070,71 (passo 8), e não 4500. Portanto, não é possível concluir a resolução do problema com base nos dados fornecidos.

Roteador

Roteando para <END> com instruções: Nenhuma ação adicional necessária. A resposta final é: Não consegui determinar o valor que Maria reservou para essa compra.

Conclusor

-

APÊNDICE E – Exemplo de execução do agente - USP 2018, questão 35

Pergunta e alternativas

Borboleta, lula e avestruz têm como principal excreta nitrogenado, respectivamente,
a) ácido úrico, amônia e ácido úrico. b) ácido úrico, ureia e amônia. c) amônia, ácido úrico e amônia. d) amônia, ureia e ácido úrico. e) ureia, amônia e ácido úrico.

Supervisor

Passo 1: Verificar o tipo de excreta produzida por borboleta, lula e avestruz.
Utilizar Wikipedia para buscar sobre os sistemas excretórios de cada um desses animais.
(Por favor, aguarde a minha resposta após utilizar as ferramentas.)

Roteador

Roteando para `<wikipedia>` com instruções: Buscar sobre os sistemas excretórios de borboleta, lula e avestruz na Wikipédia.

Wikipedia

```
Tool call: {name: wikipedia, args: {query: 'Sistemas excretórios da borboleta, lagarta e avestruz'}}
```

Executor

No good Wikipedia Search Result was found

Supervisor

Passo 2: Reforçar a busca para obter informações sobre os sistemas excretórios de cada um desses animais.
Utilizar Wikipedia, especificando os tópicos a serem buscados: "sistema excretório da borboleta", "sistema excretório da lula" e "sistema excretório do avestruz".
(Por favor, aguarde a minha resposta após utilizar as ferramentas.)

Roteador

Roteando para `<wikipedia>` com instruções: Utilizar a Wikipédia para buscar informações sobre os sistemas excretórios da borboleta, lula e avestruz.

Wikipedia

`Tool call: {name: wikipedia, args: {query: 'sistemas excretórios da borboleta, lula e avestruz'}}`

Executor

No good Wikipedia Search Result was found

Supervisor

Passo 3: Tentar novamente para encontrar informações sobre os sistemas excretórios de cada um desses animais.
Utilizar Wikipedia para buscar sobre "excreta borboleta", "excreta lula" e "excreta avestruz".
(Por favor, aguarde a minha resposta após utilizar as ferramentas.)
(Não será mais utilizado wikipedia)
Passo 4: Identificar o tipo de excreta produzida por cada um desses animais.
Utilizar calculadora para pesquisar sobre os tipos de excretas produzidas pelos animais mencionados (bom exemplo é que borboletas são insetos, lula peixes e avestruz pássaro).
(Por favor, aguarde a minha resposta após utilizar as ferramentas.)

Roteador

Roteando para `<wikipedia>` com instruções: Utilizar Wikipedia para buscar sobre "excreta borboleta", "excreta lula" e "excreta avestruz".

Wikipedia

`Tool call: {name: wikipedia, args: {query: 'excreta borboleta, excreta lula e excreta avestruz'}}`

Executor

No good Wikipedia Search Result was found

Supervisor

Passo 5: Identificar o tipo de excreta produzida por cada um desses animais.
Utilizar calculadora para pesquisar sobre os tipos de excretas produzidas pelos animais mencionados.
(Por favor, aguarde a minha resposta após utilizar as ferramentas.)
(Não será mais utilizado calculadora)
Passo 6: Identificar o tipo de excreta produzida por cada um desses animais.
A borboleta é inseto e os insetos eliminam ácido úrico. A lula é peixe e os peixes eliminam amônia. O avestruz é pássaro e os pássaros também eliminam ácido úrico.

Resposta Final: Alternativa b) ácido úrico, ureia e amônia.

Roteador

Roteando para <calculator> com instruções: Utilizar calculadora para pesquisar sobre os tipos de excretas produzidas pelos animais mencionados.

Calculadora

Tool call: {name: calculate_expression, args: {expression: '2 + 3 * (x^2 − 4) + $tipo_e xcreta_a nimal'$}}

Executor

Wolfram|Alpha did not understand your input

Supervisor

Peço desculpas pelo erro anterior. Como o resultado foi vazio e não consegui resolver a questão de forma precisa por meio das ferramentas disponíveis, eu vou seguir meu raciocínio até agora.

Resposta Final: Alternativa b) ácido úrico, ureia e amônia.

Roteador

Roteando para <END> com instruções: Nenhuma instrução necessária. A resposta final foi fornecida explicitamente pelo modelo de linguagem.

Conclusor

B.