



I<sup>2</sup>A<sup>2</sup>

institut d'intelligence  
artificielle appliquée

Agentes Autônomos - Reasoning

## Grupo 276

Diomar	didinascimento08@yahoo.com
Diego	diego@agencialothus.com.br
Rafael	rafael.calovi@meta.com.br
Leonardo	leonardo.rennerdev@gmail.com
Fagner	neofagner@gmail.com
Eduarda	eduarda.santos@meta.com.br

Agentes Autônomos - Reasoning.....	5
1. Introdução.....	5
Como os LLMs resolvem problemas complexos?.....	5
2. Chain-Of-Thought.....	6
3. Rationale Engineering .....	9
4. MCP (Model Context Protocol).....	14
5. Fully Supervised Finetuning.....	17
Como funciona?.....	18
Desvantagens / Limitações.....	18
Exemplo simples .....	18
Exemplo 1: Resolver problemas matemáticos com raciocínio estruturado .....	19
Exemplo 2: Suporte ao diagnóstico clínico com raciocínio passo a passo .....	19
Teste proposto: Avaliação da Eficácia do Ajuste Fino Supervisionado em Classificação de Texto.....	21
6. Self-Consistency Sampling.....	22
O que é Amostragem de Autoconsistência? .....	22
Como funciona?.....	23
Por que funciona melhor que respostas únicas? .....	23
Aplicações práticas.....	23
Limitações .....	24
Exemplo simplificado 1 .....	24
Exemplo 2: Problema de Idades.....	24
Exemplo 3: TESTE PRÁTICO USANDO CLAUDE.....	26
Respostas do Modelo:.....	27
Conclusão:.....	31
7. Memory and Contextual Reasoning .....	32

Liderança Inteligente: Como Memory & Contextual Reasoning Transformam Agentes Autônomos em Parceiros Estratégicos .....	32
O que é Memory and Contextual Reasoning?.....	32
Por que utilizar essa abordagem?.....	32
Comparativo: IA Tradicional x IA com Memory & Contextual Reasoning.....	34
Aplicabilidade na liderança.....	34
Estudo de Caso: Coordenadora de Operações com Suporte de IA Inteligente.....	35
Uma Proposta Visionária.....	35
8. Prompting e In-Context Learning .....	36
Prompting.....	36
In-Context Learning.....	36
Formas de In-Context Learning.....	37
Resumo da diferença entre Prompting e In-Context Learning.....	37
9. Referência.....	38

# **Agentes Autônomos - Reasoning**

## **1. Introdução**

O raciocínio em LLMs ainda tem se mostrado um grande desafio para as empresas, apesar do surgimento constante de novas técnicas e padrões sobre como "fazer a máquina pensar".

O artigo LLM Reasoning, descreve que "ainda não se sabe completamente como os modelos de IA podem aprender e aproveitar essa capacidade para resolver problemas complexos", o que abre espaço para discussões sobre quais técnicas podem ser aplicadas para melhorar ou otimizar os resultados esperados.

Dessa forma, o profissional, além de conhecer as técnicas de reasoning, precisa também desmistificar seu problema em termos lógicos, matemáticos, entre outros. Mesmo que o problema não tenha relação direta com matemática, os LLMs tendem a raciocinar melhor quando recebem uma abstração estruturada do problema. Isso significa que, ao aplicar uma técnica adequada, é possível orientar o modelo a produzir uma resposta mais eficaz.

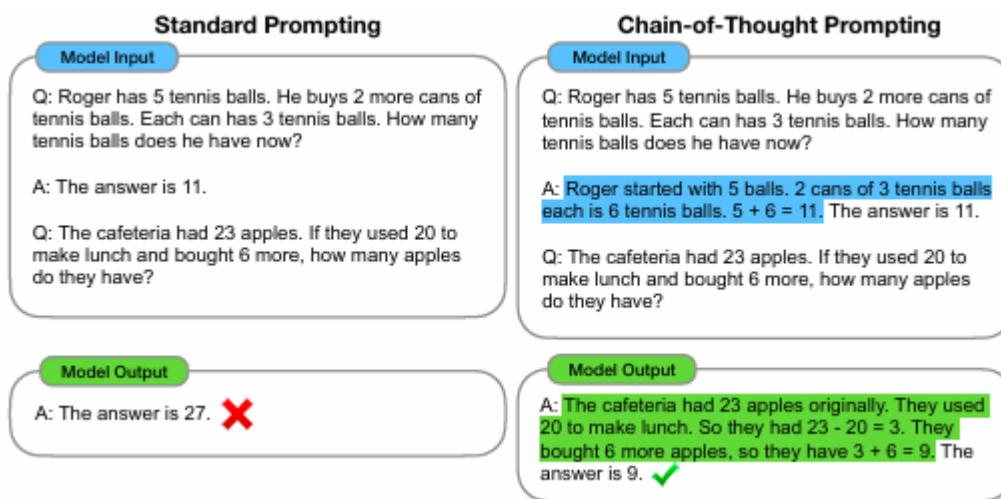
### **Como os LLMs resolvem problemas complexos?**

Para tarefas simples, como responder a perguntas diretas ou traduzir frases, os LLMs conseguem fazer isso muito bem. No entanto, quando a tarefa é mais complexa e exige um pensamento passo a passo, como resolver um problema matemático, o modelo precisa "raciocinar". Ele faz isso dividindo o problema em várias etapas, resolvendo cada parte aos poucos. Um truque interessante que os cientistas descobriram é que, se você disser algo como "vamos pensar passo a passo" para o modelo, ele pode seguir esse padrão e resolver problemas mais complicados de forma mais eficaz.

## 2. Chain-Of-Thought

A estratégia de Chain of Thought (Cadeia de Pensamento) tem como principal objetivo modelar o raciocínio de modelos de linguagem diante de um problema, por meio de deduções lógicas encadeadas. Essa abordagem consiste, essencialmente, em decompor um problema complexo em etapas menores e mais manejáveis, facilitando o processo de resolução.

Ao adotar essa técnica, os LLMs deixam de simplesmente fornecer uma resposta final, passando a explicitar os passos intermediários do raciocínio. Em outras palavras, ao lidar com uma tarefa complexa, o modelo gera uma sequência coerente de pensamentos, como se estivesse “pensando em voz alta”. Cada etapa dessa cadeia serve como base lógica para a próxima dedução, conduzindo a uma solução mais robusta e interpretável.



Essa capacidade de decomposição e autoexplicação não apenas melhora a precisão das respostas — especialmente em tarefas que envolvem múltiplas etapas de inferência —, como também facilita a interpretação e depuração do comportamento do modelo. Isso é particularmente relevante em contextos que exigem transparência e confiabilidade na tomada de decisão automatizada.

No contexto dos LLMs, o termo “inferência” refere-se ao processo pelo qual o modelo utiliza o conhecimento internalizado durante o treinamento para fazer

previsões ou gerar respostas. Após a fase de treinamento, o modelo não continua aprendendo ativamente enquanto responde aos usuários; em vez disso, ele aplica o que foi previamente aprendido para oferecer respostas imediatas. Esse processo de aplicação do conhecimento aprendido para resolver tarefas em tempo real é precisamente o que se denomina inferência.

Por exemplo, quando um usuário interage com um modelo como o ChatGPT, o que ocorre é a ativação de mecanismos internos que acessam padrões e relações linguísticas previamente assimilados, permitindo ao modelo gerar uma resposta coerente e relevante — frequentemente através de uma cadeia de pensamento estruturada, se essa estratégia for utilizada.

Essa abordagem foi sistematicamente explorada por Wei et al. (2022), que demonstraram que Chain of Thought prompting pode melhorar substancialmente o desempenho dos modelos de linguagem em tarefas de raciocínio matemático e lógico, especialmente quando comparado a abordagens tradicionais que visam respostas imediatas sem explicitação intermediária.

**Prompt:**

"Um fazendeiro tem 30 animais, entre galinhas e coelhos. No total, ele conta 84 patas. Quantas galinhas e quantos coelhos o fazendeiro tem? Pense passo a passo."

**Resposta**

"Vou resolver este problema passo a passo. Temos que determinar quantas galinhas e quantos coelhos o fazendeiro tem, sabendo que:

O total de animais é 30 (galinhas + coelhos)

O total de patas é 84

Cada galinha tem 2 patas

Cada coelho tem 4 patas

Vamos chamar de  $g$  o número de galinhas e  $c$  o número de coelhos.

Passo 1: Montar as equações com base nas informações dadas. Equação 1:  $g + c = 30$  (total de animais) Equação 2:  $2g + 4c = 84$  (total de patas)

Passo 2: Isolar uma variável na primeira equação. Da Equação 1:  $g = 30 - c$

Passo 3: Substituir na segunda equação.  $2(30 - c) + 4c = 84$   $60 - 2c + 4c = 84$   
 $60 + 2c = 84$

Passo 4: Resolver para c (número de coelhos).  $2c = 84 - 60$   $2c = 24$   $c = 12$

Passo 5: Encontrar g (número de galinhas) substituindo c na Equação 1.  $g = 30$   
 $- c$   $g = 30 - 12$   $g = 18$

Passo 6: Verificar se a solução está correta.

Total de animais:  $18 + 12 = 30 \checkmark$

Total de patas:  $(18 \times 2) + (12 \times 4) = 36 + 48 = 84 \checkmark$

Portanto, o fazendeiro tem 18 galinhas e 12 coelhos.”

### 3. Rationale Engineering

Rationale Engineering refere-se ao desenvolvimento e à aplicação de métodos que incentivam o raciocínio dos LLMs. A própria estratégia de Chain-of-Thought é uma dessas técnicas, mas não é a única. Nesse contexto, o “pensar em voz alta” também faz muito sentido — seja na hora de um prompt explicitar seu “passo a passo”, seja ao analisar uma resposta gerada pelo modelo para tentar replicar e entender o raciocínio utilizado.

**Entre as técnicas já citadas, existem outras, como:**

- Rationale Refinement – Como o nome sugere, trata-se do refinamento de exemplos de raciocínio, com o objetivo de melhor eliciar o pensamento nos LLMs.
- Rationale Exploration and Verification – Essa técnica envolve a exploração e verificação do raciocínio gerado, com o intuito de validar se o pensamento do modelo foi preciso e coerente.

Um ponto importante a ser mencionado é que, ao aplicar engenharia de raciocínio para resolver problemas complexos — como os das áreas de matemática, engenharia, biologia molecular, entre outras —, não é realista esperar que a máquina “pense” apenas com base em prompts. Não se pode depender unicamente de técnicas de escrita, já que é importante lembrar que a máquina não aprende no momento da inferência.

Para esses casos mais complexos, o ideal é analisar profundamente o problema e modelá-lo de forma adequada, pensando em como encaixar esse raciocínio de maneira comprehensível para a máquina. Ou seja, mais do que apenas usar uma técnica de prompt, é necessário estruturar a lógica do problema de forma que o modelo possa simular o raciocínio humano com maior fidelidade.

## **Prompt**

“Explique como a Revolução Industrial impactou a economia global. Antes de dar a resposta final, identifique os principais fatores envolvidos, apresente argumentos a favor e contra, e só então sintetize uma conclusão equilibrada.”

## **Resposta**

“A Revolução Industrial foi um período de profundas transformações socioeconômicas e tecnológicas que teve início na Inglaterra no século XVIII e se espalhou pelo mundo. Seu impacto na economia global foi monumental e multifacetado. Para comprehendê-lo, é fundamental analisar os principais fatores envolvidos, bem como os argumentos a favor e contra seus desdobramentos.

### **Principais Fatores Envolvidos:**

**Inovações Tecnológicas:** O desenvolvimento e a aplicação de máquinas, como a máquina a vapor, o tear mecânico e o descaroçador de algodão, foram o cerne da Revolução Industrial. Essas inovações substituíram a produção artesanal pela maquinofatura, aumentando exponencialmente a capacidade produtiva.

**Abundância de Recursos Naturais:** Países como a Inglaterra possuíam vastas reservas de carvão (fonte de energia) e ferro (matéria-prima para máquinas e infraestrutura), essenciais para o processo de industrialização.

**Acúmulo de Capital:** O capital acumulado através do comércio (incluindo o comércio transatlântico de escravos), da agricultura e das políticas mercantis protecionistas forneceu os recursos financeiros necessários para investir em fábricas, máquinas e infraestrutura.

**Mão de Obra Disponível:** As leis de cercamentos, que expulsaram camponeses de suas terras na Inglaterra, geraram um grande contingente de mão de obra desempregada e barata, disponível para trabalhar nas novas fábricas urbanas.

**Mercado Consumidor em Expansão:** O crescimento populacional e a expansão colonial criaram mercados consumidores cada vez maiores para os produtos manufaturados.

**Mentalidade Empreendedora e Liberalismo Econômico:** O desenvolvimento de uma burguesia empreendedora e a gradual ascensão do pensamento liberal, que defendia a não intervenção estatal na economia, favoreceram o ambiente para o crescimento industrial.

**Melhorias nos Transportes e Comunicações:** A construção de ferrovias, canais e o aprimoramento de navios a vapor, juntamente com inovações como o telégrafo, facilitaram o escoamento da produção e a comunicação.

**Argumentos a Favor:**

**Aumento da Produtividade e Produção em Massa:** As máquinas permitiram uma produção em larga escala, barateando os custos e tornando os bens mais acessíveis a uma parcela maior da população.

**Crescimento Econômico e Riqueza:** A industrialização impulsionou o crescimento econômico sem precedentes, gerando riqueza para os países industrializados e para a burguesia.

**Avanços Tecnológicos Contínuos:** A Revolução Industrial inaugurou um ciclo de inovações tecnológicas que continuou a impulsionar o desenvolvimento em diversos setores, como transportes, comunicações e medicina.

**Urbanização e Desenvolvimento de Cidades:** O crescimento das indústrias resultou na formação de grandes centros urbanos, que se tornaram polos de desenvolvimento e oportunidades (apesar dos problemas sociais).

**Surgimento de Novas Classes Sociais:** A burguesia industrial e o proletariado se consolidaram como classes sociais, redefinindo as estruturas sociais e políticas.

**Integração do Mercado Global:** A produção em massa e a melhoria dos transportes facilitaram o comércio internacional, interligando economias e mercados em escala global.

### **Argumentos Contra:**

Exploração do Trabalho e Condições Precárias: A busca por lucro máximo levou à exploração da mão de obra, com longas jornadas de trabalho, salários baixos, trabalho infantil e condições insalubres nas fábricas.

Intensificação das Desigualdades Sociais: A riqueza gerada pela industrialização concentrou-se nas mãos de poucos, aumentando drasticamente a disparidade entre ricos e pobres e gerando miséria em grandes centros urbanos.

Degradação Ambiental: O uso intensivo de combustíveis fósseis (carvão) e a ausência de regulamentação ambiental levaram a uma poluição severa do ar e da água, além do esgotamento de recursos naturais.

Desemprego Tecnológico: A substituição do trabalho humano pelas máquinas resultou em desemprego para muitos artesãos e trabalhadores manuais, gerando tensões sociais e movimentos de protesto (como o ludismo).

Crises de Superprodução: A capacidade de produção superava a capacidade de consumo, levando a crises econômicas periódicas de superprodução.

Dependência e Imperialismo: A busca por matérias-primas e mercados consumidores levou as potências industrializadas a expandir seu domínio sobre outras regiões do mundo, impulsionando o imperialismo e a exploração colonial.

### **Conclusão Equilibrada:**

A Revolução Industrial foi um divisor de águas na história econômica global, redefinindo a forma como as mercadorias eram produzidas, distribuídas e consumidas. Embora tenha sido um motor de crescimento econômico sem precedentes, impulsionando avanços tecnológicos e a integração de mercados, ela também gerou profundas consequências sociais e ambientais negativas.

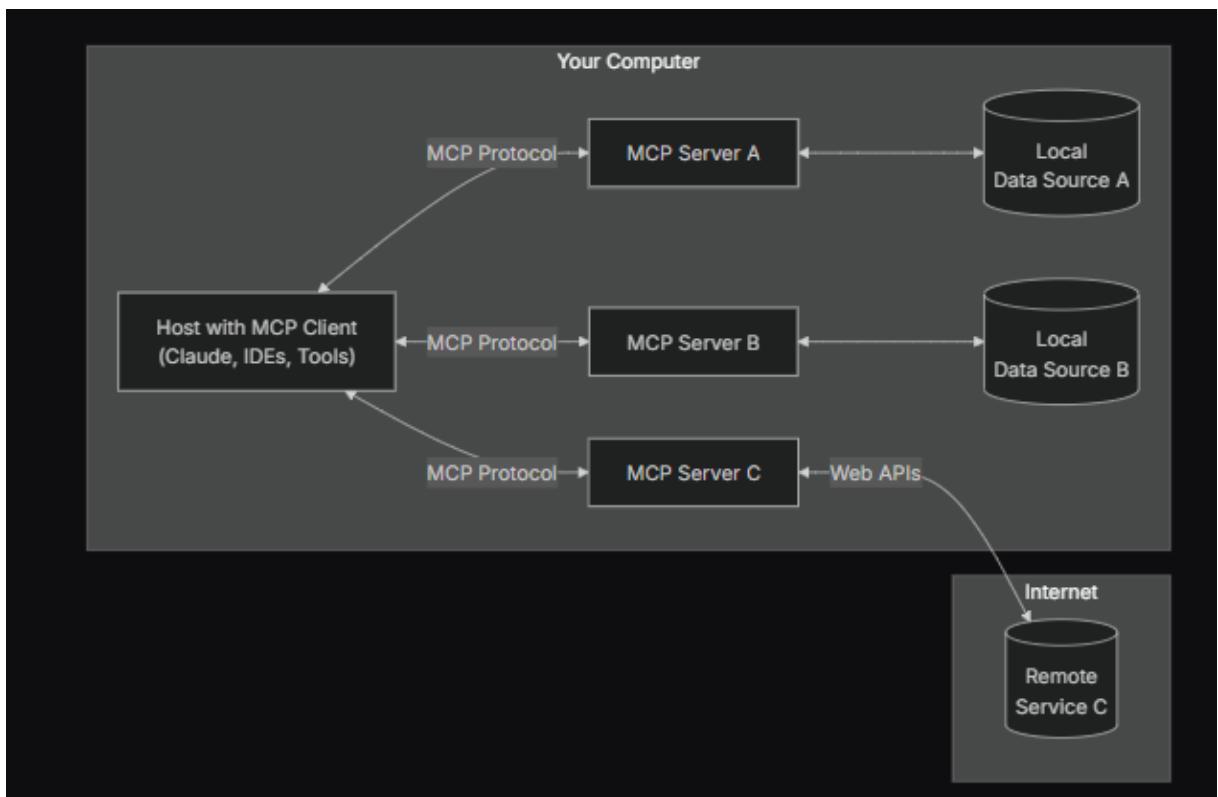
Do ponto de vista econômico, a Revolução Industrial consolidou o sistema capitalista e deu início à era da produção em massa e do consumo globalizado. Aumentou a capacidade produtiva da humanidade e diversificou os bens disponíveis.

No entanto, essa expansão veio acompanhada de uma distribuição desigual da riqueza, exploração do trabalho e impactos ambientais que perduram até hoje.

Em suma, a Revolução Industrial impactou a economia global ao transformar a produção, o comércio, as relações de trabalho e a própria organização social em escala mundial. Gerou um progresso material significativo para muitos, mas à custa de consideráveis problemas sociais e ambientais, cujas soluções e consequências continuam a ser desafios para a economia global contemporânea.”

## 4. MCP (Model Context Protocol)

O MCP é um protocolo cliente-servidor que visa simplificar e aprimorar a comunicação entre diferentes componentes de sistemas que interagem com LLMs. Ele foi projetado para atuar como uma ponte, facilitando a troca de dados e a integração entre diversas fontes de dados e ferramentas.



Uma das principais vantagens do MCP é a sua integração pré-construída. Isso significa que ele já oferece conexões prontas para uso com uma variedade de plataformas e serviços, o que reduz significativamente o tempo e o esforço necessários para configurar a comunicação. Além disso, o protocolo é flexível, permitindo que você alterne facilmente entre diferentes provedores de LLMs ou fontes de dados, sem a necessidade de reestruturar todo o seu sistema. Essa adaptabilidade é crucial em um ambiente onde as tecnologias e as necessidades podem mudar rapidamente.

A segurança também é um pilar fundamental do MCP. Ele foi desenvolvido com foco em garantir que a comunicação seja robusta e segura, protegendo as informações

trocadas entre os componentes e as LLMs. Assim como outros protocolos de comunicação, o MCP opera na arquitetura cliente-servidor, onde um cliente (a aplicação que precisa interagir com a LLM ou dados) faz requisições a um servidor (que gerencia a comunicação e as integrações).

Imagine sua equipe de desenvolvimento como um grupo de chefs em uma cozinha movimentada, e o MCP é o sistema de comunicação que conecta tudo, garantindo que os ingredientes certos cheguem aos lugares certos no momento certo.

Imagine sua equipe de desenvolvimento como um grupo de chefs em uma cozinha movimentada, e o MCP é o sistema de comunicação que conecta tudo, garantindo que os ingredientes certos cheguem aos lugares certos no momento certo.

- Front-end (O Garçom): É o garçom que recebe os pedidos dos clientes (usuários) e os leva para a cozinha. Ele não sabe como preparar o prato, apenas anota o que o cliente quer.
- API (O Maître): O maître é o cérebro da cozinha. Ele traduz os pedidos do garçom em instruções para os cozinheiros, gerencia os ingredientes e coordena a entrega dos pratos. Ele sabe onde cada ingrediente está e qual cozinheiro faz o quê.
- Banco de Dados (A Despensa de Ingredientes): É a despensa onde todos os ingredientes frescos e preparados são armazenados de forma organizada. O maître sabe exatamente o que está na despensa e onde encontrar.
- LLM (O Chef Especialista): O chef especialista é aquele que consegue criar pratos complexos e personalizados com base em um vasto conhecimento culinário (os dados e informações). Ele pode, por exemplo, criar uma receita de bolo "quase um pão" se você pedir.
- RAG (Retrieval Augmented Generation - O Livro de Receitas Inteligente): O RAG é como um livro de receitas muito esperto. Quando o chef especialista (LLM) precisa de uma informação específica para um prato, ele consulta esse livro que não só tem muitas receitas, mas também sabe onde buscar novas receitas se não tiver a que precisa. Pense em um livro de receitas que pode magicamente expandir seu conteúdo quando você pergunta algo que ele não conhece.

O MCP é como o sistema de intercomunicação unificado da cozinha. Ele permite que o maître (API) se comunique facilmente com a despensa (banco de dados), o livro de receitas inteligente (RAG) e o chef especialista (LLM), sem ter que aprender a "linguagem" específica de cada um. Ele simplifica, flexibiliza e torna mais

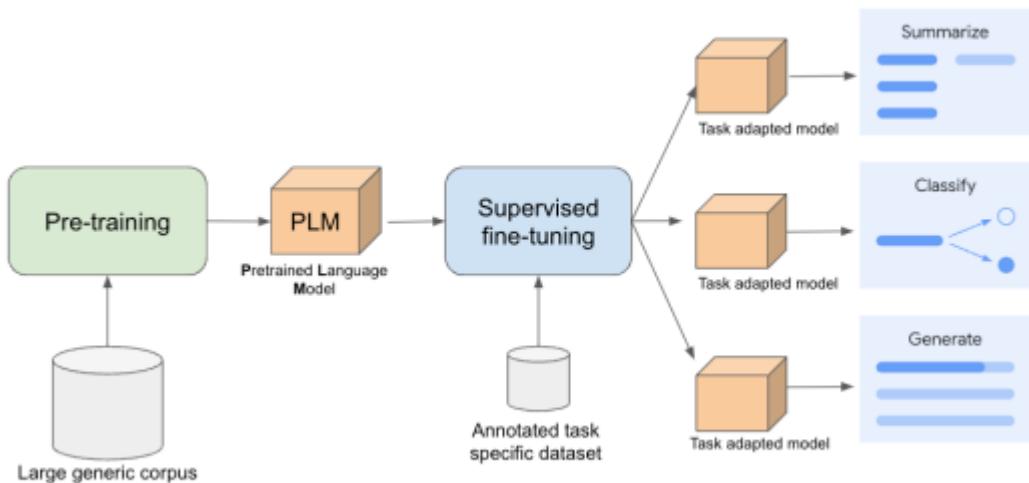
segura a forma como todos esses "chefs" trabalham juntos para servir o melhor prato (a melhor resposta ou funcionalidade) ao cliente.

## 5. Fully Supervised Finetuning

O Ajuste Fino Totalmente Supervisionado é uma técnica utilizada para aprimorar a capacidade de um modelo de linguagem na realização de tarefas específicas com maior precisão. Esse processo orienta o modelo a seguir um raciocínio mais correto e confiável, garantindo respostas estruturadas e coerentes.

A abordagem consiste em ajustar um modelo previamente treinado, ou seja, que já possui um vasto conhecimento, utilizando exemplos específicos e detalhados, conhecidos como dados rotulados. Esses exemplos fornecem diretrizes claras para que o modelo refine sua compreensão e aprimore sua capacidade de raciocinar de maneira explícita e estruturada.

Essa técnica permite que os modelos de linguagem "pensem em voz alta", tornando suas respostas mais transparentes e confiáveis. No entanto, a qualidade e a clareza das demonstrações de raciocínio contidas no conjunto de dados são elementos cruciais para o sucesso do ajuste fino. Quanto mais bem definidos forem os exemplos fornecidos, melhor será a capacidade do modelo de aplicar um pensamento lógico e preciso na execução de suas tarefas.



## Como funciona?

1. **Uso de dados rotulados:** São fornecidos exemplos claros em que tanto a entrada (por exemplo, uma pergunta) quanto a saída correta (por exemplo, a resposta esperada com raciocínio) estão bem definidas. Esses pares servem como guia para o modelo.
2. **Aprendizado supervisionado:** O modelo compara o que ele responde com a resposta correta nos dados. Se houver erro, os ajustes são feitos nos seus parâmetros internos para se aproximar mais da resposta desejada. Esse processo é repetido milhares (ou milhões) de vezes.
3. **Objetivo:** Ensinar o modelo a seguir um caminho de raciocínio mais preciso e a dar respostas mais confiáveis em tarefas específicas, como resolver problemas matemáticos, responder perguntas complexas, ou tomar decisões lógicas.

## Desvantagens / Limitações

1. **Requer dados com raciocínio explícito:** O modelo só aprende bem se os dados de treino mostrarem claramente os passos do raciocínio, o que nem sempre é fácil de conseguir. Esses dados precisam ser feitos ou anotados manualmente, o que dá trabalho.
2. **Fica limitado ao conjunto de dados usado:** Como o ajuste é feito com um conjunto específico de exemplos, o modelo pode ficar “viciado” naquele tipo de pergunta ou naquele estilo de raciocínio. Isso pode reduzir sua flexibilidade para lidar com outros contextos.

## Exemplo simples

Imagine que você quer que o modelo aprenda a explicar por que uma afirmação é verdadeira em lógica. Em vez de apenas mostrar a pergunta e a resposta final, você dá:

- **Entrada:** “Se todos os humanos são mortais, e Sócrates é humano, Sócrates é mortal. Explique.”
- **Saída:** “Todos os humanos são mortais (premissa 1). Sócrates é humano (premissa 2). Logo, Sócrates é mortal (conclusão lógica).”

Esses dados com raciocínio passo a passo são usados para ensinar o modelo a fazer o mesmo tipo de explicação no futuro.

## **Exemplo 1: Resolver problemas matemáticos com raciocínio estruturado**

Desafio: Modelos de linguagem geralmente cometem erros ao resolver problemas de matemática se não forem ensinados a seguir uma sequência lógica de passos.

### **Solução com Ajuste Fino Supervisionado:**

1. **Criação do conjunto de dados rotulado:** Prepara-se um conjunto com milhares de perguntas matemáticas, onde cada pergunta vem acompanhada da solução explicada passo a passo. **Exemplo de dado de treino:**
  - a. **Entrada (pergunta):** “Se João tem 3 vezes mais maçãs que Maria, e Maria tem 4 maçãs, quantas maçãs João tem?”
  - b. **Saída (resposta com raciocínio):** “Maria tem 4 maçãs. João tem 3 vezes isso:  $3 \times 4 = 12$ . Portanto, João tem 12 maçãs.”
2. **Ajuste do modelo com esse conjunto de dados:** O modelo é treinado comparando suas respostas com essas soluções corretas. Se ele errar um passo, ele é ajustado para tentar seguir o mesmo caminho lógico da resposta correta na próxima vez.
3. **Resultado:** Depois do ajuste fino, o modelo passa a resolver esse tipo de problema de forma mais precisa, explicando o raciocínio de forma clara, em vez de simplesmente “chutar” a resposta.

Impacto: Esse tipo de ajuste já foi usado em benchmarks como o GSM8K (conjunto de dados de problemas matemáticos de nível fundamental), e levou modelos como o GPT a melhorar muito na capacidade de mostrar raciocínio estruturado, chegando a acertar mais de 90% das questões.

## **Exemplo 2: Suporte ao diagnóstico clínico com raciocínio passo a passo**

Desafio: Modelos de linguagem podem fornecer sugestões de diagnóstico baseadas em sintomas, mas sem raciocínio estruturado, correm o risco de errar ou

parecer inseguros. Na medicina, isso é inaceitável — é essencial explicar o porquê de cada possível diagnóstico.

### **Solução com Ajuste Fino Supervisionado:**

1. **Criação de dados rotulados com raciocínio médico explícito:** Usa-se um conjunto de dados contendo casos clínicos reais ou simulados, onde cada um apresenta:
  - a. **Descrição dos sintomas e histórico do paciente (entrada)**
  - b. **Hipótese diagnóstica correta, com explicação médica passo a passo (saída) Exemplo de dado de treino:**
  - c. **Entrada (caso clínico):** “Paciente de 45 anos com dor no peito em aperto, irradiando para o braço esquerdo, iniciada há 30 minutos, associado a sudorese.”
  - d. **Saída (resposta com raciocínio):** “Dor em aperto, duração prolongada e irradiação para o braço são sinais típicos de angina instável ou infarto agudo do miocárdio. A sudorese reforça a possibilidade de um quadro isquêmico agudo. O diagnóstico mais provável é Infarto Agudo do Miocárdio.”
2. **Ajuste fino do modelo:** O modelo é treinado para imitar esse raciocínio clínico em novos casos, aprendendo como médicos pensam ao avaliar sintomas.
3. **Resultado:** O modelo agora consegue:
  - a. Sugerir diagnósticos com justificativa clínica clara
  - b. Evitar suposições apressadas ou sem base lógica
  - c. Ajudar médicos como ferramenta de apoio (nunca substituindo o profissional)

**Impacto:** Esse tipo de treinamento tem sido explorado em contextos como:

- Triagem automatizada em plataformas de saúde digital
- Sistemas de apoio à decisão médica em hospitais
- Chatbots médicos supervisionados, que respondem com base em protocolos clínicos

## **Teste proposto: Avaliação da Eficácia do Ajuste Fino Supervisionado em Classificação de Texto**

Objetivo: Avaliar o impacto do ajuste fino totalmente supervisionado na performance de um modelo pré-treinado de linguagem (ex: BERT) em uma tarefa específica de classificação de sentimentos (positivo, negativo, neutro).

1. **Base de Dados:** Utilizar o conjunto de dados IMDb ou SST-2 (Stanford Sentiment Treebank), que contêm frases e suas respectivas classificações sentimentais.
2. **Etapas do Teste:** a. **Modelo Pré-treinado:** Utilize um modelo como o bert-base-uncased, pré-treinado em tarefas gerais de linguagem. b. **Cenários a comparar:** \* **Cenário 1 (sem ajuste fino):** O modelo pré-treinado é usado diretamente, sem nenhuma adaptação à base de dados-alvo. \* **Cenário 2 (com ajuste fino supervisionado):** O mesmo modelo é ajustado (fine-tuned) usando os dados rotulados da tarefa. c. **Métricas de Avaliação:** \* Acurácia \* F1-score \* Matriz de confusão
3. **Hipótese:** O modelo ajustado com dados supervisionados deve apresentar ganhos significativos de desempenho em relação ao modelo sem ajuste fino.
4. **Resultados Esperados (exemplo simulado):**

<b>Cenário</b>	<b>Acurácia</b>	<b>F1-Score</b>
Sem ajuste fino	65%	0.60
Com ajuste fino supervisionado	90%	0.89

Exportar para as Planilhas

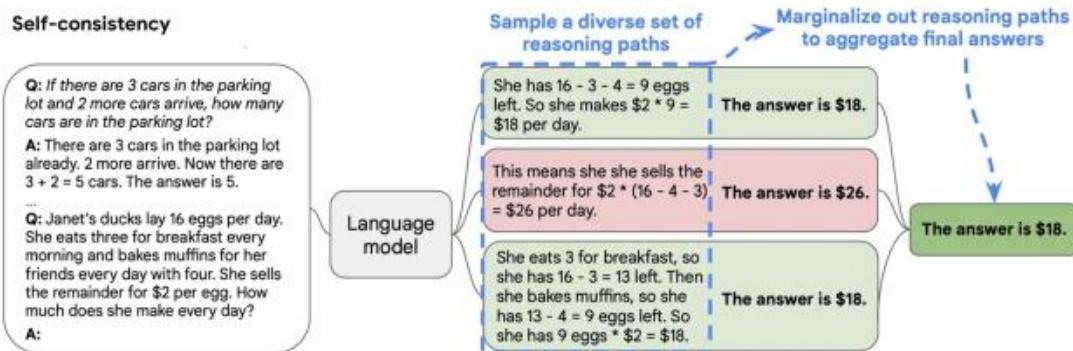
Conclusão Esperada: O ajuste fino totalmente supervisionado permite que o modelo generalista se especialize na tarefa desejada, o que leva a melhorias significativas nos resultados.

## 6. Self-Consistency Sampling

A autoconsistência é uma abordagem que simplesmente pergunta a um modelo a mesma prompt várias vezes e leva o resultado da maioria das respostas como resposta final. É uma continuação da abordagem da Cadeia de Pensamento (CdP ou CoT:Chain of Thought, em inglês) e é ainda mais poderosa quando usada em conjunto.

A Amostragem de Autoconsistência é uma técnica poderosa para melhorar o desempenho dos modelos de linguagem grande (LLMs) em tarefas que exigem raciocínio complexo, como resolver problemas matemáticos, lógicos ou realizar inferências.

Em vez de gerar uma única resposta a partir de uma pergunta, o modelo é instruído a gerar várias respostas diferentes ao mesmo tempo. Em seguida, analisa-se o conjunto dessas respostas para verificar qual delas aparece com mais frequência ou qual segue uma linha de raciocínio mais coerente com as demais. A resposta mais comum ou mais consistente é então escolhida como resposta final.



### O que é Amostragem de Autoconsistência?

É uma técnica que simula como os humanos abordam problemas difíceis: gerando múltiplos caminhos de raciocínio independentes para a mesma questão e depois identificando a resposta mais consistente ou frequente entre eles.

Quando uma pergunta envolve várias etapas de raciocínio, um modelo pode cometer erros em uma ou outra etapa, especialmente se for forçado a responder de

forma direta. No entanto, se for solicitado a responder a mesma pergunta várias vezes com pequenas variações internas, ele pode explorar múltiplos caminhos de pensamento.

## Como funciona?

1. **Geração múltipla:** O mesmo problema é apresentado ao modelo várias vezes (geralmente 5-100 vezes).
2. **Diversidade de raciocínio:** Em cada tentativa, o modelo é configurado com diferentes parâmetros de aleatoriedade (temperatura, top\_p, etc.), produzindo diversos caminhos de raciocínio.
3. **Agregação de respostas:** As múltiplas respostas são coletadas e analisadas para identificar padrões.
4. **Escolha por maioria:** A resposta que aparece com mais frequência é selecionada como a resposta final.

## Por que funciona melhor que respostas únicas?

1. **Compensação de erros:** Mesmo quando um modelo ocasionalmente comete erros, a resposta correta tende a aparecer mais frequentemente nas múltiplas tentativas.
2. **Exploração de possibilidades:** Permite ao modelo explorar diferentes abordagens para um problema, simulando a forma como humanos consideram várias perspectivas.
3. **Redução de vieses:** Ajuda a diminuir vieses ou "fixações" em caminhos de raciocínio específicos que podem levar a erros.
4. **Maior confiabilidade:** Fornece uma medida implícita de confiança - quanto mais respostas convergem para o mesmo resultado, maior a probabilidade de estar correto.

## Aplicações práticas

- **Problemas matemáticos:** Particularmente eficaz para cálculos multietapa onde um único erro pode comprometer todo o resultado.
- **Raciocínio lógico:** Melhora significativamente o desempenho em quebra-cabeças ou deduções complexas.
- **Programação:** Pode gerar múltiplas soluções para um problema de código e escolher a mais funcional ou eficiente.

- **Tradução complexa:** Em traduções que requerem interpretação contextual sofisticada.

## Limitações

- **Custo computacional:** Exige muito mais recursos computacionais, pois cada problema é resolvido múltiplas vezes.
- **Tempo de processamento:** Não é ideal para aplicações em tempo real que exigem respostas imediatas.
- **Incerteza persistente:** Se nenhuma resposta clara surgir como maioria, a técnica pode não conseguir resolver a ambiguidade.

## Exemplo simplificado 1

Imagine um problema de matemática: "Se João tem 12 maçãs e dá 1/3 para Maria, quantas maçãs João tem agora?"

Um modelo pode gerar estas respostas em diferentes tentativas:

- **Caminho 1:** "João tem 12 maçãs. Dá 1/3 para Maria, ou seja,  $12 \div 3 = 4$  maçãs. Então João fica com  $12 - 4 = 8$  maçãs."
- **Caminho 2:** "João começa com 12 maçãs.  $1/3$  de 12 = 4. Depois de dar 4 para Maria, João tem 8 maçãs."
- **Caminho 3:** "João tem 12 maçãs.  $1/3$  delas é  $12 \times (1/3) = 4$ . João fica com  $12 - 4 = 8$  maçãs."
- **Caminho 4:** "João tem 12 maçãs. Ele dá 1/3, ou seja,  $12 \times 0,33 = 3,96$ , arredondando para 4. João fica com  $12 - 4 = 8$  maçãs."
- **Caminho 5:** "12 maçãs no total. João dá 1/3 para Maria.  $1/3$  de 12 = 4. João fica com  $12 - 4 = 8$  maçãs."

A resposta "8 maçãs" aparece consistentemente, logo é selecionada como resposta final.

## Exemplo 2: Problema de Idades

Pergunta: "Pedro tem o dobro da idade de Ana. Juntos, eles têm 36 anos. Quantos anos tem cada um?"

**O modelo gera várias respostas:**

### **Caminho 1:**

Pedro tem o dobro da idade de Ana  $\rightarrow$  Pedro = 2A

$$\text{Pedro} + \text{Ana} = 36 \rightarrow 2A + A = 36 \rightarrow 3A = 36 \rightarrow A = 12$$

$$\text{Pedro} = 24, \text{Ana} = 12$$

### **Caminho 2:**

Se Ana tem x anos, então Pedro tem 2x.

$$x + 2x = 36 \rightarrow 3x = 36 \rightarrow x = 12 \rightarrow \text{Pedro} = 24, \text{Ana} = 12$$

### **Caminho 3:**

Pedro tem o dobro de Ana  $\rightarrow$  Pedro = 2 × Ana

$$\text{Soma} = 36 \rightarrow 2 \times \text{Ana} + \text{Ana} = 36 \rightarrow 3 \times \text{Ana} = 36 \rightarrow \text{Ana} = 12, \text{Pedro} = 24$$

Caminho 4 (com um pequeno erro):

Pedro é 2 vezes mais velho que Ana  $\rightarrow$  Pedro = Ana + Ana = 14 + 14 = 28?

Ana = 14, Pedro = 28  $\rightarrow$  Soma = 42 (⚠ erro)

### **Caminho 5:**

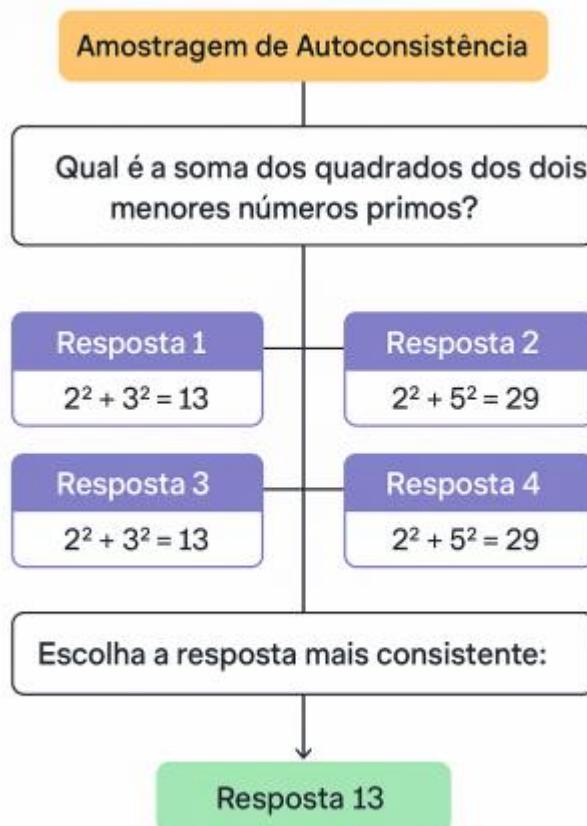
Pedro = 2 × Ana

Pedro + Ana = 36

$$2A + A = 36 \rightarrow 3A = 36 \rightarrow A = 12 \rightarrow \text{Pedro} = 24, \text{Ana} = 12$$

#### **✓ Resultado:**

- **Respostas geradas:**
  - [Pedro: 24, Ana: 12]  $\rightarrow$  aparece 4 vezes
  - [Pedro: 28, Ana: 14]  $\rightarrow$  aparece 1 vez (incorrecto)
- **Resposta final por autoconsistência:** ✓ Pedro tem 24 anos, Ana tem 12 anos  
A autoconsistência funciona como uma "sabedoria coletiva" do próprio modelo, permitindo que ele chegue a conclusões mais robustas através de múltiplas tentativas de resolução do mesmo problema.



### Exemplo 3: TESTE PRÁTICO USANDO CLAUDE

**Teste com uma pergunta matemática simples como "Qual é o resultado de 17 × 24?"**

Para garantir que o modelo não esteja simplesmente repetindo a mesma resposta memorizada, adicionamos pequenas variações ao prompt que não afetam o problema em si:

#### Prompt 1

Resolva este problema matemático passo a passo: Qual é o resultado da multiplicação de 17 × 24? Mostre cada etapa do seu raciocínio antes de fornecer a resposta final.

#### Prompt 2

Por favor, calcule 17 × 24 mostrando todo o processo. Explique cada passo do seu cálculo e apresente a resposta final.

**Prompt 3**

Quanto é 17 multiplicado por 24? Apresente sua solução de forma detalhada, descrevendo cada etapa até chegar ao resultado.

**Prompt 4**

Resolva a seguinte multiplicação:  $17 \times 24$ . Demonstre o método que você usou, descrevendo o processo passo a passo.

**Prompt 5**

Calcule o produto de 17 e 24. Mostre o procedimento completo, incluindo todos os passos intermediários.

**Prompt 6**

Se multiplicarmos 17 por 24, qual será o resultado? Detalhe cada passo do seu cálculo antes de dar a resposta final.

**Prompt 7**

Encontre o valor de  $17 \times 24$ . Explique seu método de resolução, demonstrando cada etapa do processo.

**Prompt 8**

Qual é o produto quando multiplicamos 17 por 24? Apresente sua solução passo a passo, incluindo qualquer cálculo intermediário.

**Prompt 9**

Calcule  $17 \times 24$  usando o método de sua preferência. Mostre detalhadamente como você chegou ao resultado final.

**Prompt 10**

Determine o resultado de 17 vezes 24. Explique cada passo do seu raciocínio até chegar à resposta final.

**Respostas do Modelo:****Resposta ao Prompt 1**

Para resolver  $17 \times 24$ , vou dividir em passos:

1. Primeiro, multiplico  $17 \times 4 = 68$
2. Depois, multiplico  $17 \times 20 = 340$
3. Finalmente, somo os resultados:  $68 + 340 = 408$  Portanto,  $17 \times 24 = 408$

### Resposta ao Prompt 2

Calculando  $17 \times 24$ :

Posso usar o método tradicional:

24

$\times 17$

----

28 ( $7 \times 4$ )

70 ( $7 \times 10$ )

40 ( $10 \times 4$ )

- 300 ( $10 \times 30$ ) ---- 408 Resposta final: 408

### Resposta ao Prompt 3

17 multiplicado por 24:

Vou multiplicar 17 por 24 usando decomposição:

$$17 \times 24 = 17 \times (20 + 4)$$

$$= (17 \times 20) + (17 \times 4)$$

$$= 340 + 68$$

$$= 408$$

### Resposta ao Prompt 4

Resolução da multiplicação  $17 \times 24$ :

Método: multiplicação por partes

Passo 1:  $7 \times 4 = 28$

Passo 2:  $7 \times 20 = 140$

Passo 3:  $10 \times 4 = 40$

Passo 4:  $10 \times 20 = 200$

Passo 5: Somar todos os resultados:  $28 + 140 + 40 + 200 = 408$

O resultado é 408.

### **Resposta ao Prompt 5**

Para calcular o produto de 17 e 24: Utilizando o algoritmo convencional:  $1\ 7 \times 2\ 4$

$$\begin{array}{r} 6\ 8 \\ (17 \times 4) \quad 3\ 4\ 0 \\ (17 \times 20) \end{array}$$

$$\begin{array}{r} 4\ 0\ 8 \end{array}$$

O produto de 17 e 24 é 408.

### **Resposta ao Prompt 6**

Multiplicando 17 por 24: Vou fazer de duas formas para verificação: Método 1:  $17 \times 24 = 17 \times (20 + 4) = 17 \times 20 + 17 \times 4 = 340 + 68 = 408$  Método 2:  $17 \times 24$

$$\begin{array}{r} 6\ 8 \\ (17 \times 4) \quad 3\ 4\ 0 \\ (17 \times 20) \end{array}$$

$$\begin{array}{r} 408 \end{array}$$

O resultado é 408.

### **Resposta ao Prompt 7**

Para encontrar o valor de  $17 \times 24$ :

Podemos calcular como:

- $17 \times 4 = 68$
- $17 \times 20 = 340$
- Total =  $68 + 340 = 408$  Portanto,  $17 \times 24 = 408$

### **Resposta ao Prompt 8**

O produto quando multiplicamos 17 por 24:

Utilizando a multiplicação posicional:

- Multiplico  $7 \times 4 = 28$
- Multiplico  $1 \times 4 = 4$ , que na posição das dezenas = 40
- Multiplico  $7 \times 2 = 14$ , que na posição das dezenas = 140
- Multiplico  $1 \times 2 = 2$ , que na posição das centenas = 200
- Somando:  $28 + 40 + 140 + 200 = 408$  O produto é 408.

### **Resposta ao Prompt 9**

Calculando  $17 \times 24$ :

Método escolhido: decomposição em fatores convenientes

$$17 \times 24 = 17 \times 6 \times 4$$

$$= 102 \times 4$$

$$= 408$$

Verifiquei com outro método:

$$17 \times 24 = (10 + 7) \times 24$$

$$= 10 \times 24 + 7 \times 24$$

$$= 240 + 168$$

$$= 408$$

Resultado final: 408

### **Resposta ao Prompt 10**

Para determinar 17 vezes 24: Usando o algoritmo clássico:  $17 \times 24$

68 340

408

Portanto, 17 vezes 24 é igual a 408.

## **Conclusão:**

O resultado foi unânime de 408, pois foi uma pergunta simples, mas fica exemplificado como fazer o uso da amostragem de autoconsistência para comparar e definir pela maioria dos resultados iguais.

## 7. Memory and Contextual Reasoning

### Liderança Inteligente: Como Memory & Contextual Reasoning Transformam Agentes Autônomos em Parceiros Estratégicos

A Inteligência Artificial (IA) por si só já é uma revolução. No entanto, sem memória e compreensão de contexto, ela se torna apenas uma máquina que responde, sem realmente entender. É nesse ponto que entra a estratégia de Memory and Contextual Reasoning (Memória e Raciocínio Contextual), uma abordagem que transforma agentes autônomos em verdadeiros aliados estratégicos — capazes de aprender, adaptar-se e agir com inteligência emocional simulada.

#### O que é Memory and Contextual Reasoning?

Essa estratégia se baseia em duas engrenagens centrais:

- **Memory:** Capacidade do agente de reter e utilizar informações anteriores, incluindo interações passadas, preferências do usuário e aprendizados anteriores.
- **Contextual Reasoning:** Habilidade de analisar o ambiente e o momento atual para adaptar respostas com coerência, empatia e relevância.

Combinadas, essas competências permitem à IA lembrar o que foi dito ontem e compreender o que acontece agora, atuando de forma semelhante a um ser humano atento e inteligente.

#### Por que utilizar essa abordagem?

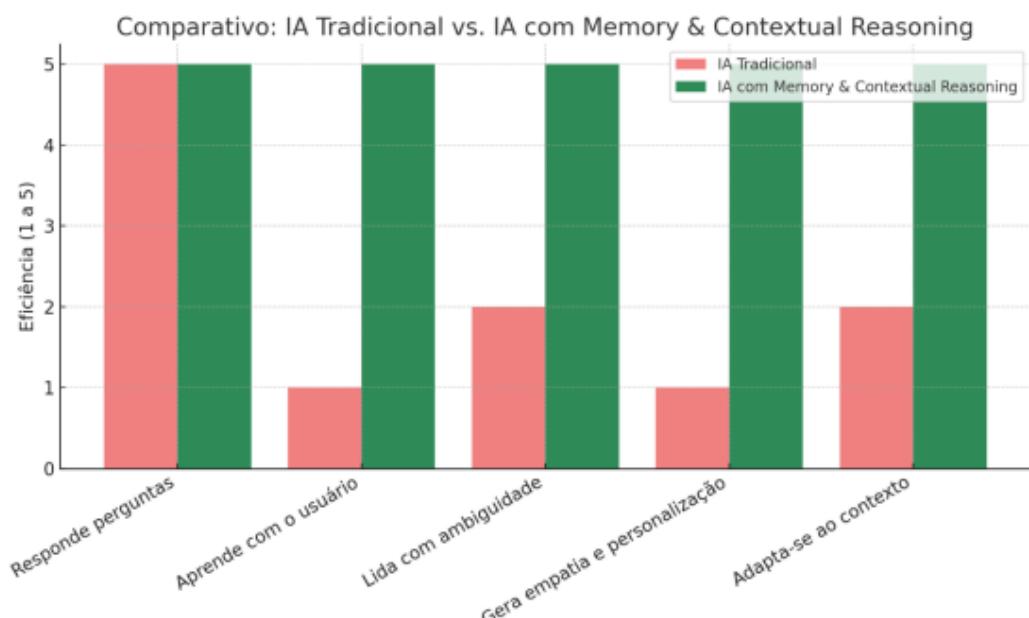
Sem Memory & Contextual Reasoning, a IA funciona como uma simples calculadora, exigindo comandos diretos e sem aprendizado contínuo. Já com essa estratégia, há melhorias significativas:

- **Maior precisão nas decisões:** Redução de erros em até **35%**, segundo estudos da OpenAI (2024).
- **Melhoria na satisfação do usuário:** **42%** a mais de interações classificadas como úteis.

- **Menor necessidade de intervenção humana:** Redução de até **60%** em cenários de atendimento inteligente.

## Comparativo: IA Tradicional x IA com Memory & Contextual Reasoning

Funcionalidade	IA Tradicional	IA com Memory & Contextual Reasoning
Responde perguntas	✓	✓
Aprende com o usuário	✗	✓
Lida com ambiguidade	⚠️ Com dificuldade	✓ Com eficiência
Gera empatia e personalização	✗	✓
Adapta-se ao contexto	⚠️ Parcialmente	✓ Totalmente



## Aplicabilidade na liderança

Para líderes, a diferença é clara: uma IA sem memória equivale a um colaborador que esquece tudo após cada reunião. Já uma IA com Memory &

Contextual Reasoning age como um verdadeiro braço direito, entendendo o histórico, analisando o contexto e oferecendo soluções personalizadas.

## **Estudo de Caso: Coordenadora de Operações com Suporte de IA Inteligente**

Duda, uma coordenadora de operações, enfrenta desafios como produtividade, qualidade e gestão de processos. A aplicação de agentes autônomos com memória e raciocínio contextual pode trazer benefícios como:

- **Redução de retrabalho:** Sistemas que aprendem com erros anteriores e ajustam procedimentos automaticamente.
- **Feedback personalizado:** IA que analisa indicadores para sugerir abordagens individuais para cada colaborador.
- **Prevenção de falhas:** O sistema pode identificar padrões e agir proativamente para evitar problemas.
- **Melhora na experiência do cliente:** Interações mais humanas, rápidas e empáticas elevam a satisfação dos consumidores.

## **Uma Proposta Visionária**

Agentes autônomos não substituem líderes — eles os potencializam. Funcionam como extensões da nossa memória e intuição, permitindo que líderes se concentrem na estratégia, na cultura e nas pessoas.

"Memória e raciocínio contextual são o novo cérebro corporativo. Quem souber usá-los com sabedoria, lidera!"

## 8. Prompting e In-Context Learning

### Prompting

#### O que é?

Prompting é o ato de enviar uma instrução ou pergunta a um modelo de linguagem para guiá-lo na geração de uma resposta. Ele funciona como o input (entrada) para a IA. Objetivo: Obter uma resposta útil e relevante do modelo, seja para responder perguntas, gerar textos, classificar dados, resumir informações etc. Exemplo simples de prompting Prompt: "Explique o conceito de energia cinética em termos simples." Resposta esperada do LLM: "É a energia que um objeto tem por estar em movimento." Engenharia de Prompts É a prática de formular prompts de maneira estratégica para obter respostas mais corretas, completas ou específicas. Exemplo de engenharia de prompt: "Explique o conceito de energia cinética em termos simples, com um exemplo do cotidiano e em no máximo 3 frases."

### In-Context Learning

#### O que é?

É a habilidade de um LLM aprender no momento da interação, usando exemplos embutidos no próprio prompt para entender e realizar uma tarefa sem precisar de novo treinamento.

#### Vantagem

- Não requer modificar os parâmetros do modelo.
- Permite adaptar o modelo a novas tarefas usando só exemplos no prompt.

## Formas de In-Context Learning

Tipo	O que é fornecido no prompt?	Exemplo Prático
<b>Zero-shot</b>	Apenas a instrução da tarefa.	<b>“Traduza para o francês:</b> Bom dia.”
<b>One-shot</b>	Um exemplo da tarefa + a instrução.	<b>“Exemplo:</b> Hello → Olá. <b>Agora traduza:</b> Good morning →”
<b>Few-shot</b>	Vários exemplos da tarefa antes da instrução final.	<b>“Exemplo 1:</b> Cat → Gato. <b>Exemplo 2:</b> Dog → Cachorro. <b>Agora traduza:</b> Bird →”

## Resumo da diferença entre Prompting e In-Context Learning

Prompting	In-Context Learning
Pode ser uma simples pergunta	Sempre envolve uma tarefa a ser aprendida
Não exige exemplos necessariamente	Pode usar exemplos como parte da entrada
Guia o modelo	<b>Ensina o modelo temporariamente</b> (durante a inferência)

## 9. Referência

WEI, Jason et al. Chain of Thought Prompting Elicits Reasoning in Large Language Models. *arXiv preprint arXiv:2201.11903*, 2022. Disponível em: <https://arxiv.org/pdf/2201.11903.pdf>. Acesso em: 21 maio 2025.

PROMPT ENGINEERING GUIDE. *LLM Reasoning*. Disponível em: <https://www.promptingguide.ai/research/llm-reasoning>. Acesso em: 21 maio 2025.

Fu, Y., Wang, X., Kang, D., & Lin, Z. (2022). Complex Rationalization Improves CoT Reasoning. Disponível em: <https://arxiv.org/abs/2210.00720>

IBM. **Fine-tuning: o que é e como funciona?** IBM Think Blog Brasil, 2024. Disponível em: <https://www.ibm.com/br-pt/think/topics/fine-tuning>. Acesso em: 21 maio 2025.

GOOGLE CLOUD. **Supervised fine-tuning for Gemini: advancing model performance with your data**. Google Cloud Blog, 2024. Disponível em: <https://cloud.google.com/blog/products/ai-machine-learning/supervised-fine-tuning-for-gemini-llm>. Acesso em: 21 maio 2025.