



Natthan Elias.

CC - 7 semestre - IFSul.

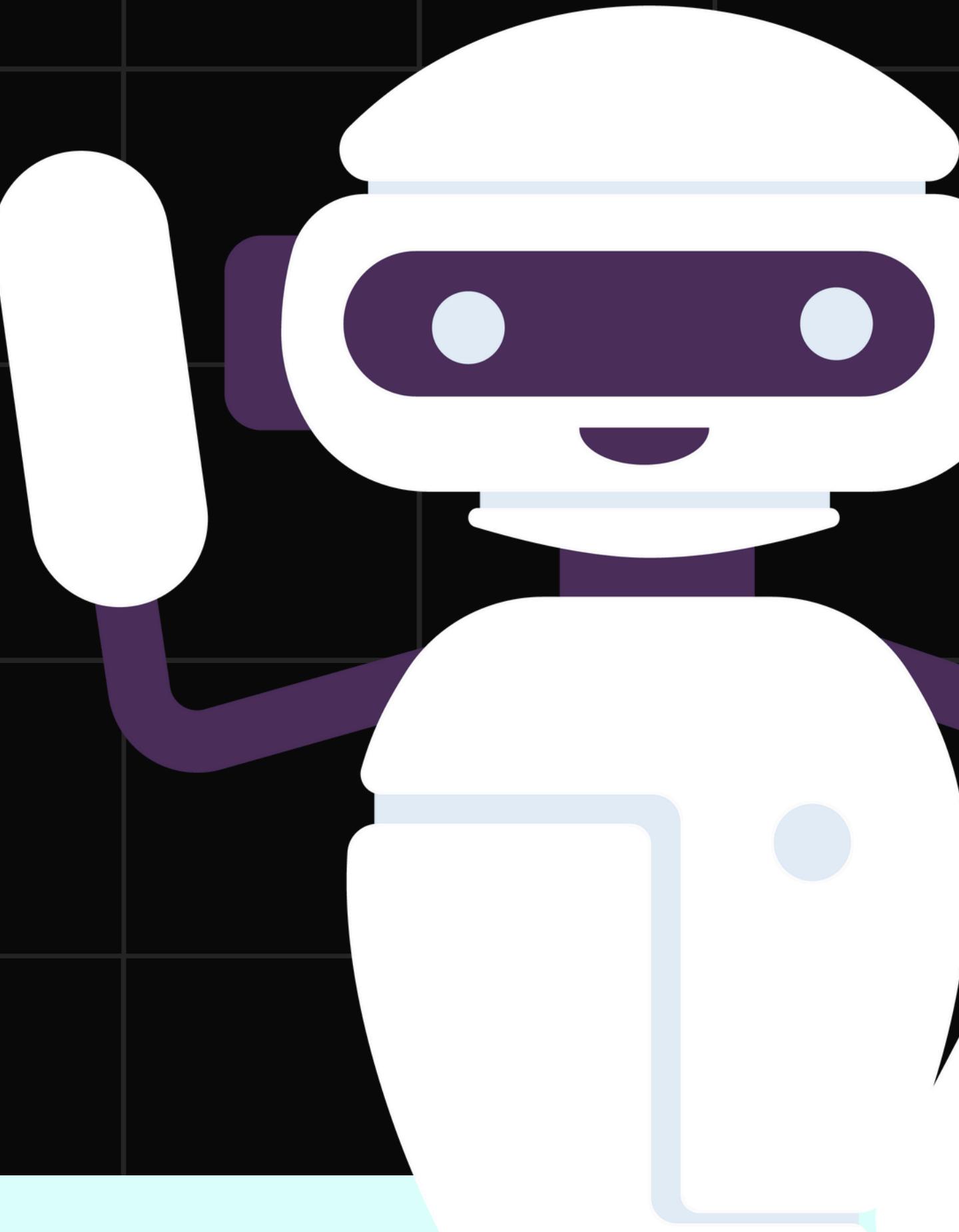
AI Engineer - Compass.UOL

Técnicas de IA Generativa

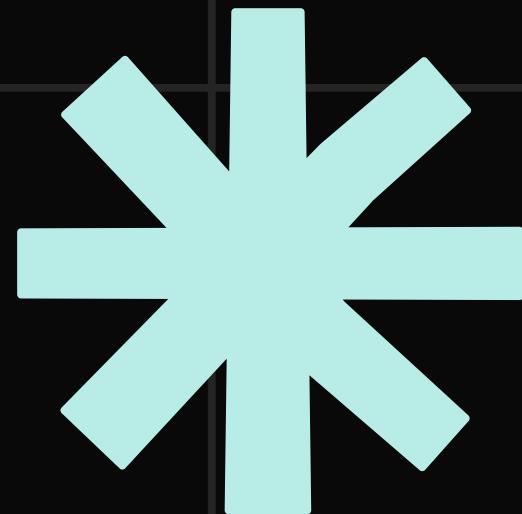
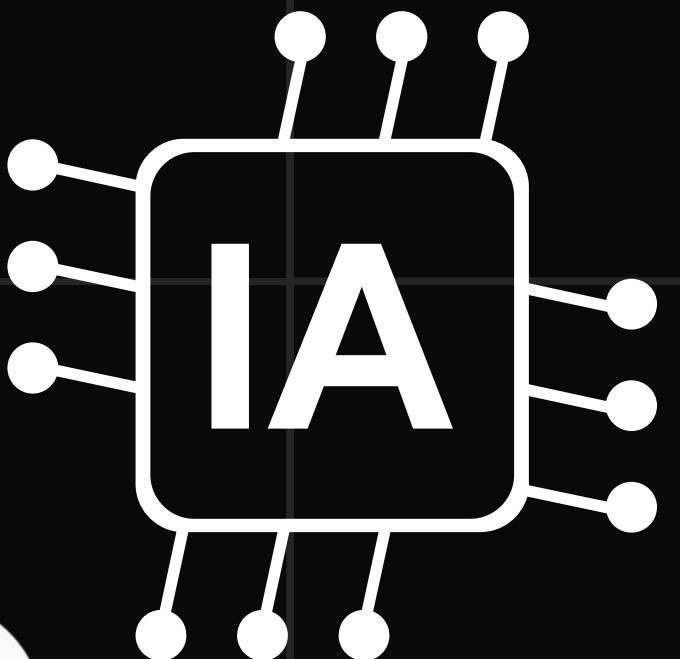
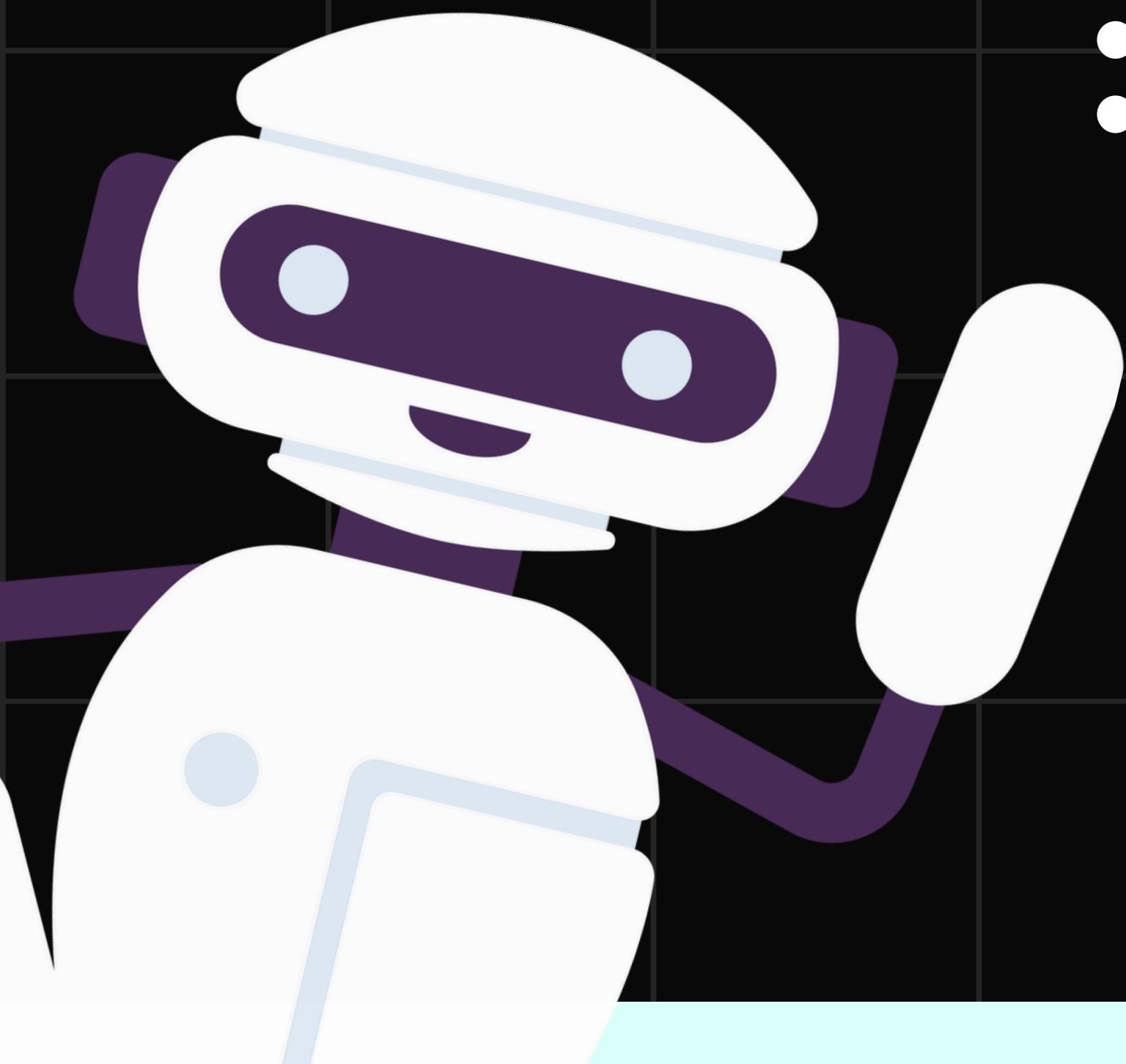
COM FOCO EM AGENTES

* ÍNDICE

- 1.0 que é IA Generativa?
- 2.0 que são Agentes?
- 3.0 que são LLMs?
- 4. Técnicas de AI Engineering
- 5. Agentes de LLMs
- 6. MCP
- 7. A2A Protocol
- 8. Frameworks para criação
de Agentes
- 9. Casos de Uso

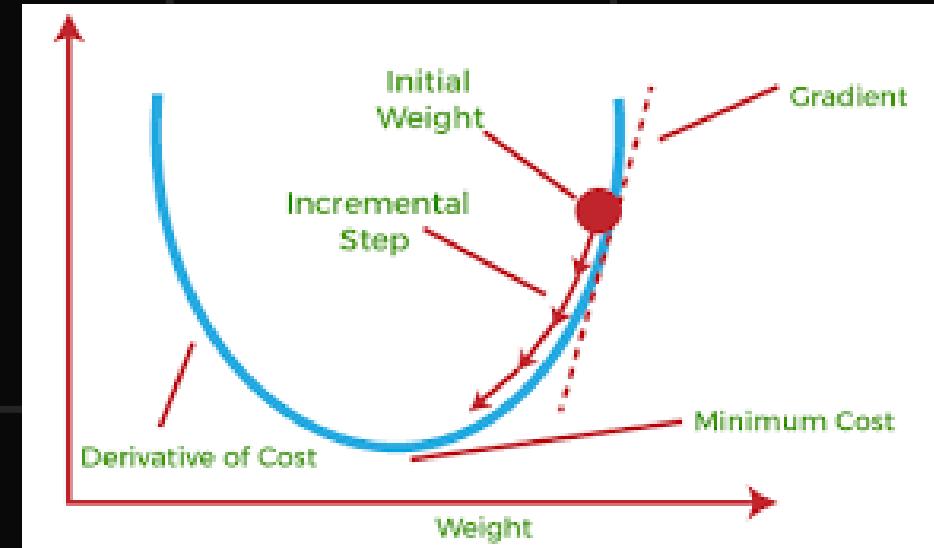
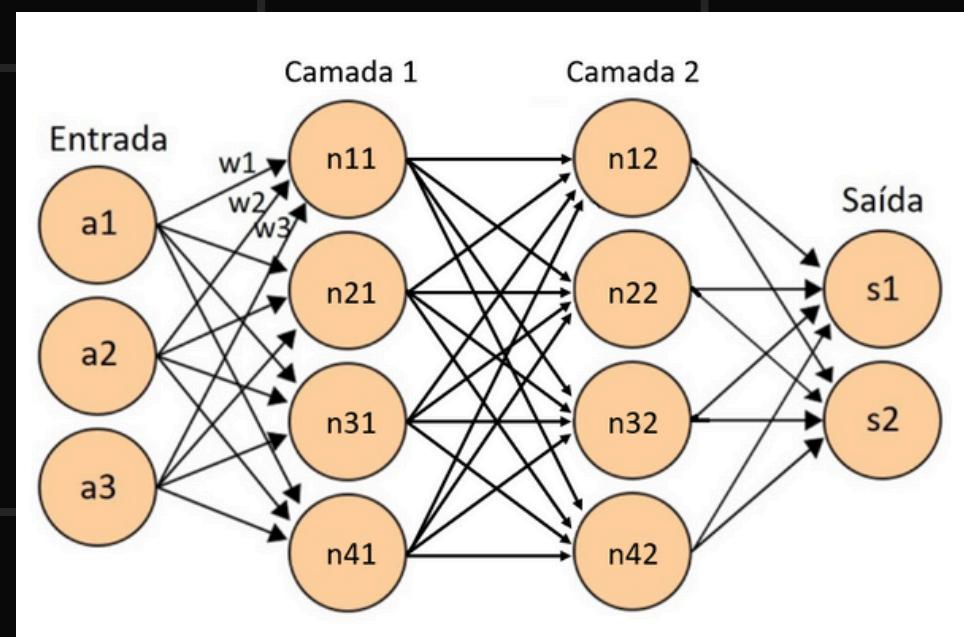
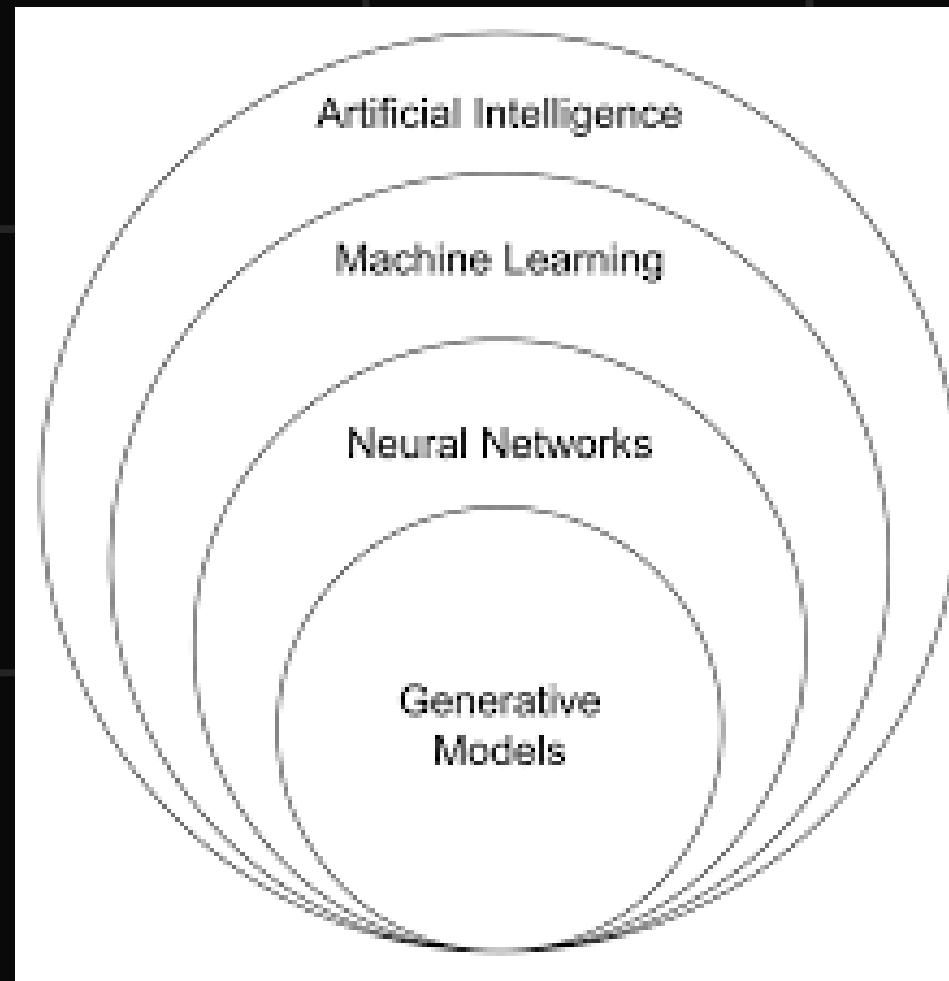


* O que é IA Generativa?



O que é IA Generativa?

- IA que é capaz de **criar conteúdos** que se assemelham a dados gerados por humanos.
- **O que a torna possível? (Big data + técnicas)**



- As RNAs são sistemas computacionais **capazes de aprender a partir de dados e identificar padrões complexos**.

IA Generativa

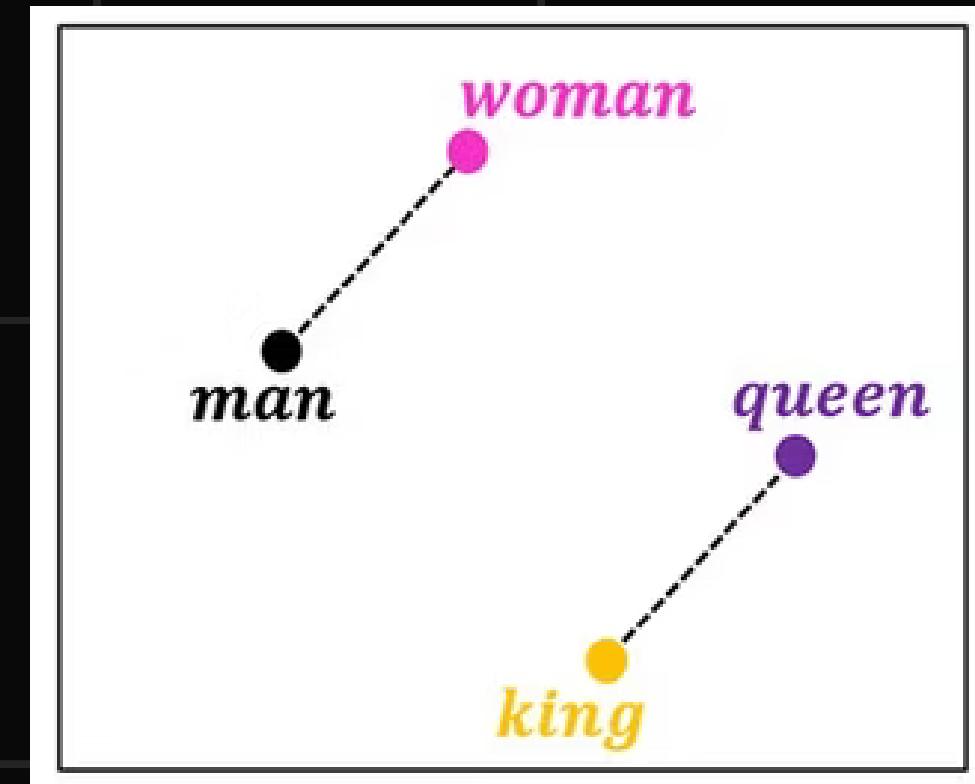
PROCESSAMENTO DE LINGUAGEM NATURAL

- Área da Linguística que emprega técnicas computacionais (ML) para **entender e gerar linguagem humana**.
- **Principais Tarefas:** Tradução automática, classificação de sentenças, análise de sentimento, sumarização de texto, etc.
- **Principal Desafio:**
- Computadores não processam informações da mesma forma que os humanos.
- **EX:** dadas duas frases como "Estou com fome" e "Estou triste", somos capazes de determinar facilmente o quanto semelhantes elas são.
 - Para modelos ML tais tarefas são mais difíceis.
- O texto precisa ser processado de uma forma que permita ao modelo aprender com ele.

IA Generativa

PROCESSAMENTO DE LINGUAGEM NATURAL

- Word Embeddings:
 - Forma de **representar palavras como vetores numéricos.**
 - Busca capturar as **relações semânticas e contextos** de um dado vocabulário.



- Palavras são representadas como vetores de alta dimensão.

$man \rightarrow$	0.6	-0.2	0.8	0.9	-0.1	-0.9	-0.7
$woman \rightarrow$	0.7	0.3	0.9	-0.7	0.1	-0.5	-0.4
$king \rightarrow$	0.5	-0.4	0.7	0.8	0.9	-0.7	-0.6
$queen \rightarrow$	0.8	-0.1	0.8	-0.9	0.8	-0.5	-0.9

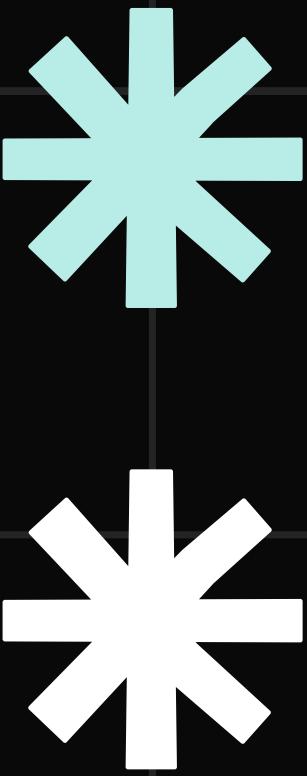
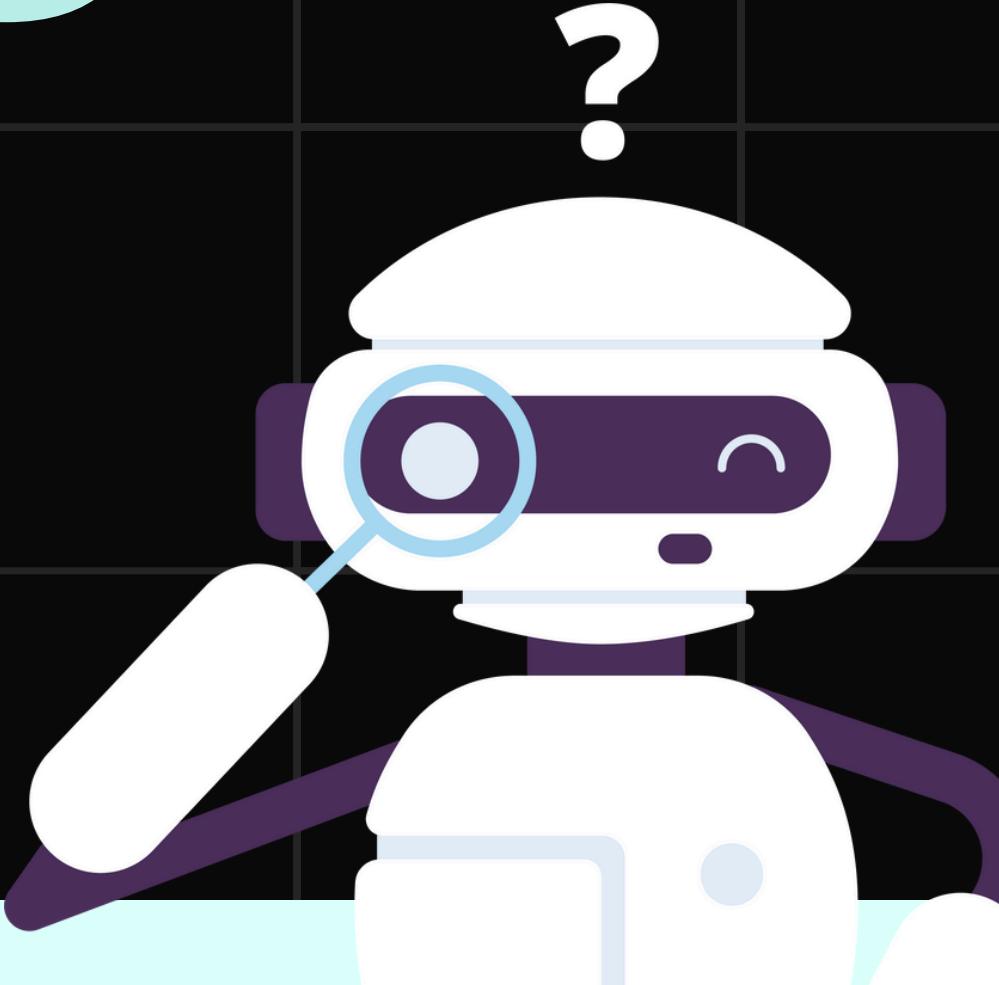
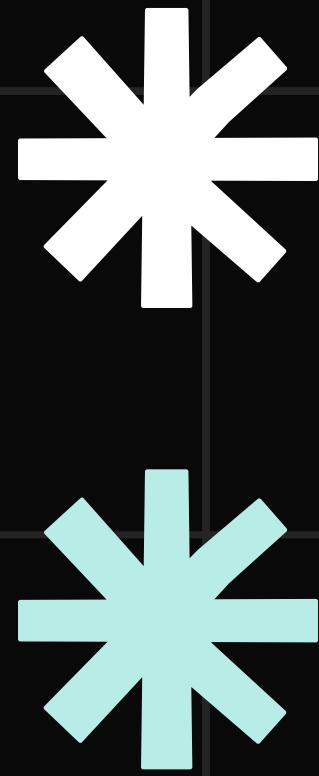
- Significado semântico e relacionamentos entre palavras.

IA Generativa

CASOS DE USO

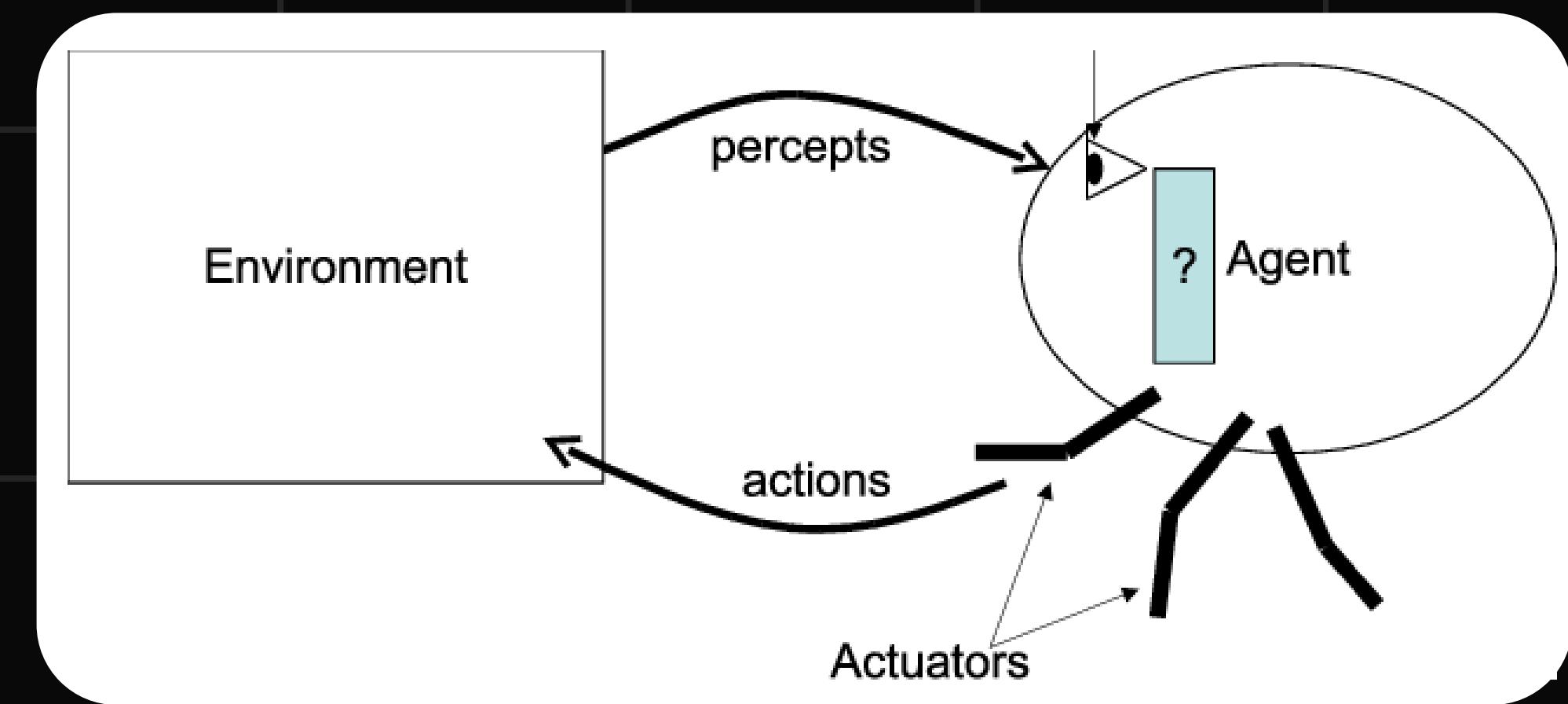
- **Codificação**
- **Produção de Imagem e Vídeo**
- **Escrita:**
 - Criação de e-mails, posts para redes sociais, blogs, textos de marketing (copywriting), relatórios e documentos.
- **Educação:**
 - Tutoria personalizada e correção de redações.
 - Integração e treinamento de funcionários.
- **Bots Conversacionais:**
 - Assistentes de IA para suporte ao cliente e companheiros interativos.
- **Resumos**
 - "Conversar" com documentos para extrair insights.
- **Organização de Dados:**
 - Processamento e organização automatizada de documentos.
- **Automação de Fluxo de Trabalho**
- ***Criatividade é o limite ...***

que são Agentes



O que são Agentes?

- **Definição Clássica:** "qualquer coisa que pode ser vista como **percebendo seu ambiente** através de sensores e **agindo sobre esse ambiente** através de atuadores"
- (Russel et al.)
- Interagem através de um ciclo contínuo de **Percepção-Ação-Feedback**.



O que são Agentes?

TIPOS CLÁSSICOS (antes dos LLMs)

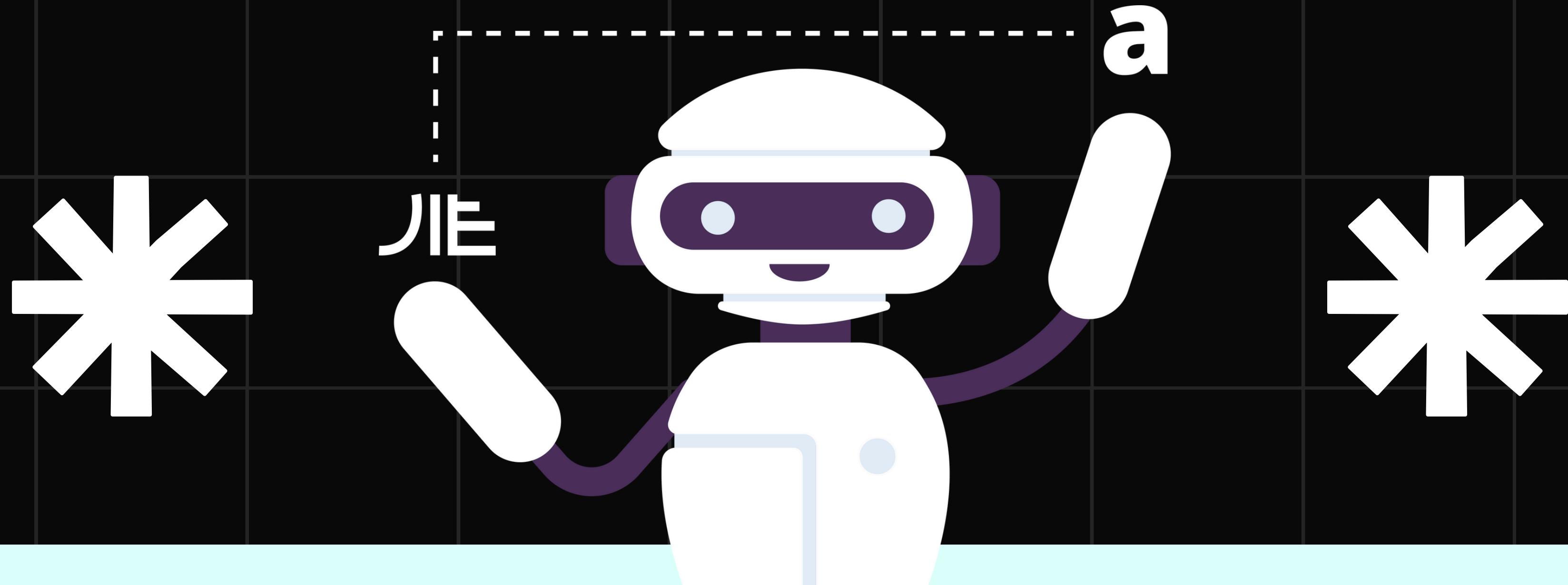
- **Regras pré-definidas:** Agentes seguem regras lógicas ou condicionais explícitas para reagir a situações.
 - **Exemplo:** autômatos celulares que geram comportamentos complexos a partir de regras simples.
- **Equações simbólicas:** Utilizam equações matemáticas (como diferenciais ou algébricas) para modelar o comportamento dos agentes.
- **Modelagem estocástica:** Introduz aleatoriedade nas decisões dos agentes para representar incertezas do mundo real.
 - **Exemplo:** modelos de escolha discreta usados para simular variações no comportamento humano.
- **Modelos de aprendizado de máquina:** Permitem que agentes aprendam com dados ou interação com o ambiente.

O que são Agentes?

CARACTERÍSTICAS

- **Autonomia (*Agency*):** Significa que o agente pode **decidir e operar sem intervenção** humana direta ou de outros agentes, e tem controle sobre suas próprias ações e **estado interno**.
- **Reatividade:** Devem ser capazes de **perceber seu ambiente** e **responder a mudanças** que ocorrem nele.
- **Sociabilidade:** Refere-se à capacidade de um agente interagir com outros agentes através de mecanismos como **cooperação, coordenação e negociação**. No mínimo, isso implica a **capacidade de se comunicar**.

O que são LLMs?



O que são LLMs ?

- **Grandes Modelos de Linguagem**
 - Treinados em uma quantidade **massiva de dados** de textos de humanos.
 - Podem **entender e gerar mídias** (texto, imagens, audio e vídeo).
 - **Simulam** pensamento encadeado de humanos.
- À medida que aumentam, demonstram capacidades que não foram explicitamente programadas ou antecipadas.

Attention Is All You Need

Ashish Vaswani*
Google Brain
avaswani@google.com

Noam Shazeer*
Google Brain
noam@google.com

Niki Parmar*
Google Research
nikip@google.com

Jakob Uszkoreit*
Google Research
usz@google.com

Llion Jones*
Google Research
llion@google.com

Aidan N. Gomez* †
University of Toronto
aidan@cs.toronto.edu

Lukasz Kaiser*
Google Brain
lukaszkaiser@google.com

Illia Polosukhin* ‡
illia.polosukhin@gmail.com

Abstract

The dominant sequence transduction models are based on complex recurrent or convolutional neural networks that include an encoder and a decoder. The best performing models also connect the encoder and decoder through an attention mechanism. We propose a new simple network architecture, the Transformer, based solely on attention mechanisms, dispensing with recurrence and convolutions entirely. Experiments on two machine translation tasks show these models to be superior in quality while being more parallelizable and requiring significantly less time to train. Our model achieves 28.4 BLEU on the WMT 2014 English-to-German translation task, improving over the existing best results, including ensembles, by over 2 BLEU. On the WMT 2014 English-to-French translation task, our model establishes a new single-model state-of-the-art BLEU score of 41.8 after training for 3.5 days on eight GPUs, a small fraction of the training costs of the best models from the literature. We show that the Transformer generalizes well to other tasks by applying it successfully to English constituency parsing both with large and limited training data.

<https://arxiv.org/pdf/1706.03762>

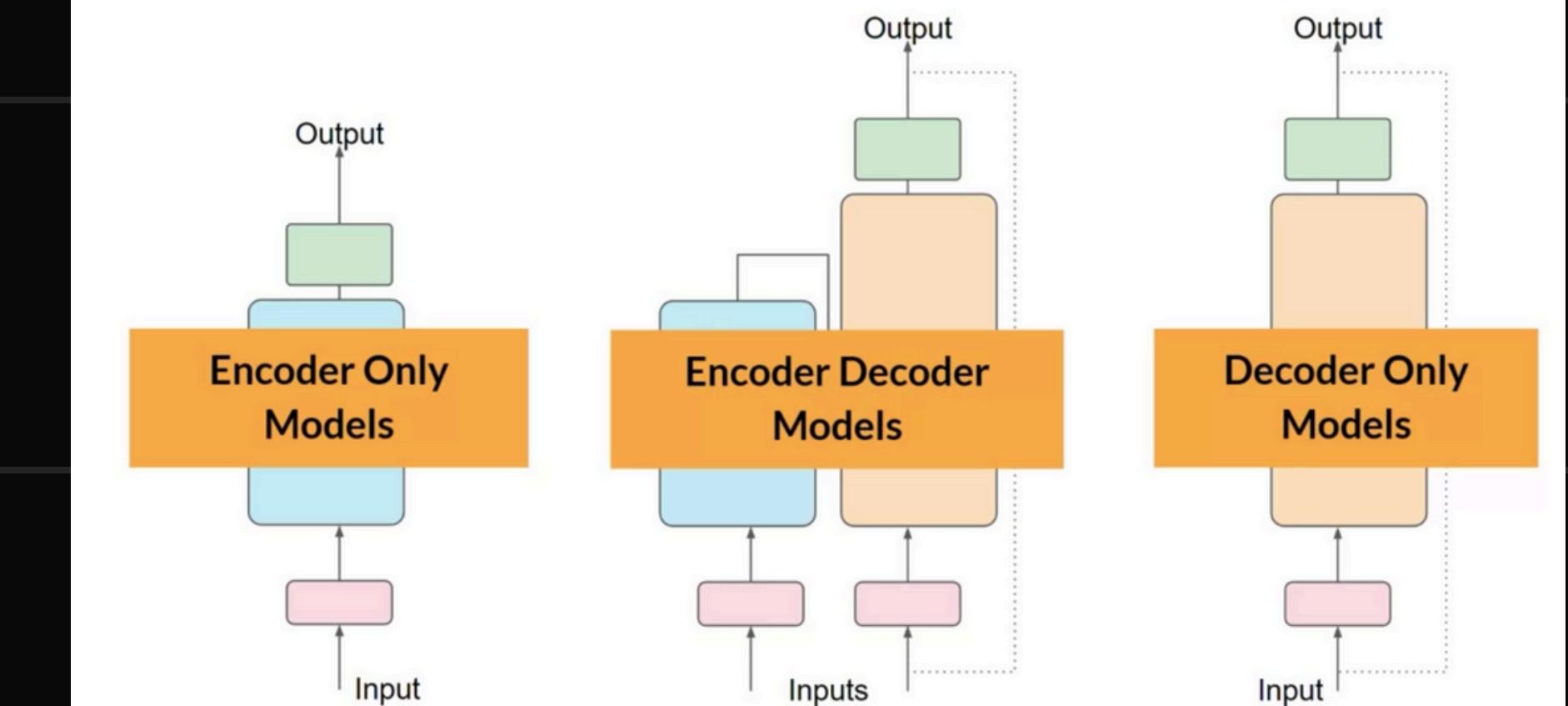
• Transformer

- **Capturar a semântica + Possibilita treinar modelos em paralelo.**

O que são LLMs ?

ARQUITETURA TRANSFORMER

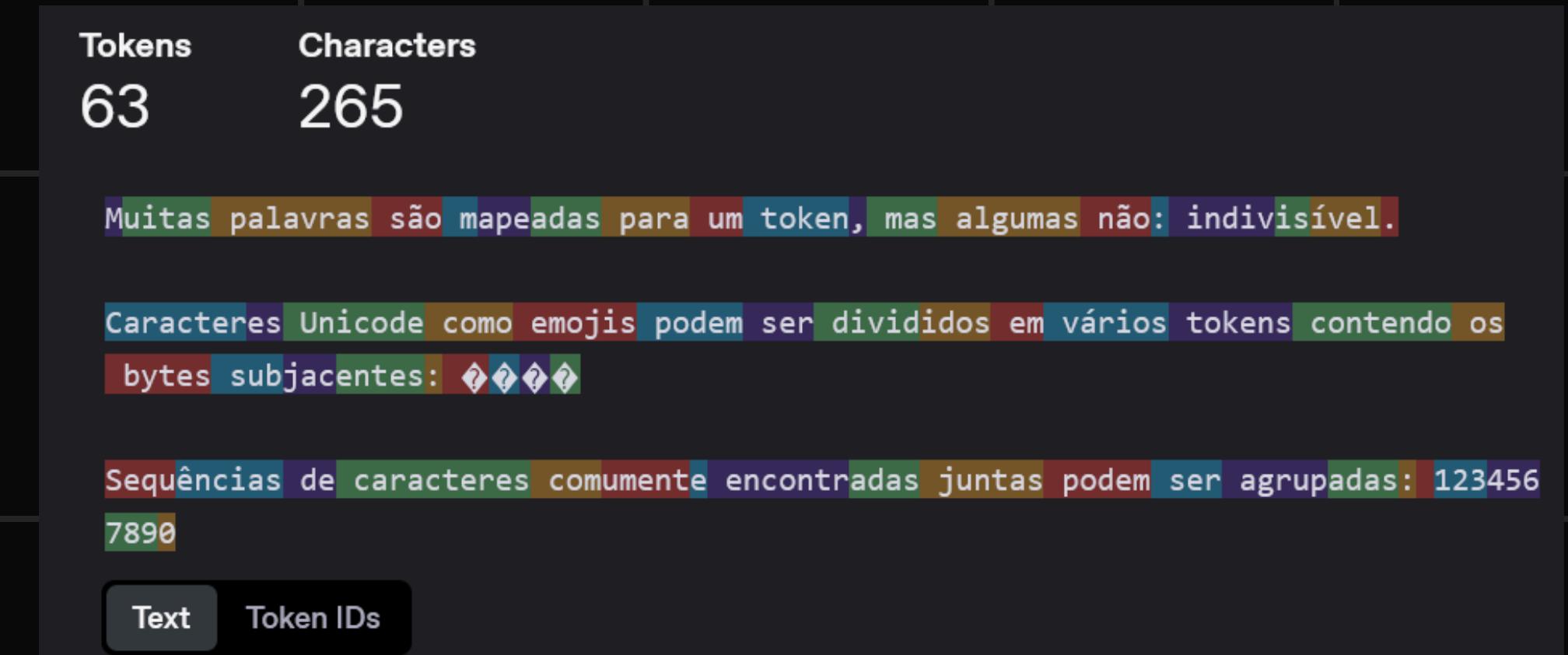
Model	Examples	Tasks
Encoder-only	BERT, DistilBERT, ModernBERT	Sentence classification, named entity recognition, extractive question answering
Decoder-only	GPT, LLaMA, Gemma, SmolLM	Text generation, conversational AI, creative writing
Encoder-decoder	BART, T5, Marian, mBART	Summarization, translation, generative question answering



O que são LLMs ?

TOKENS

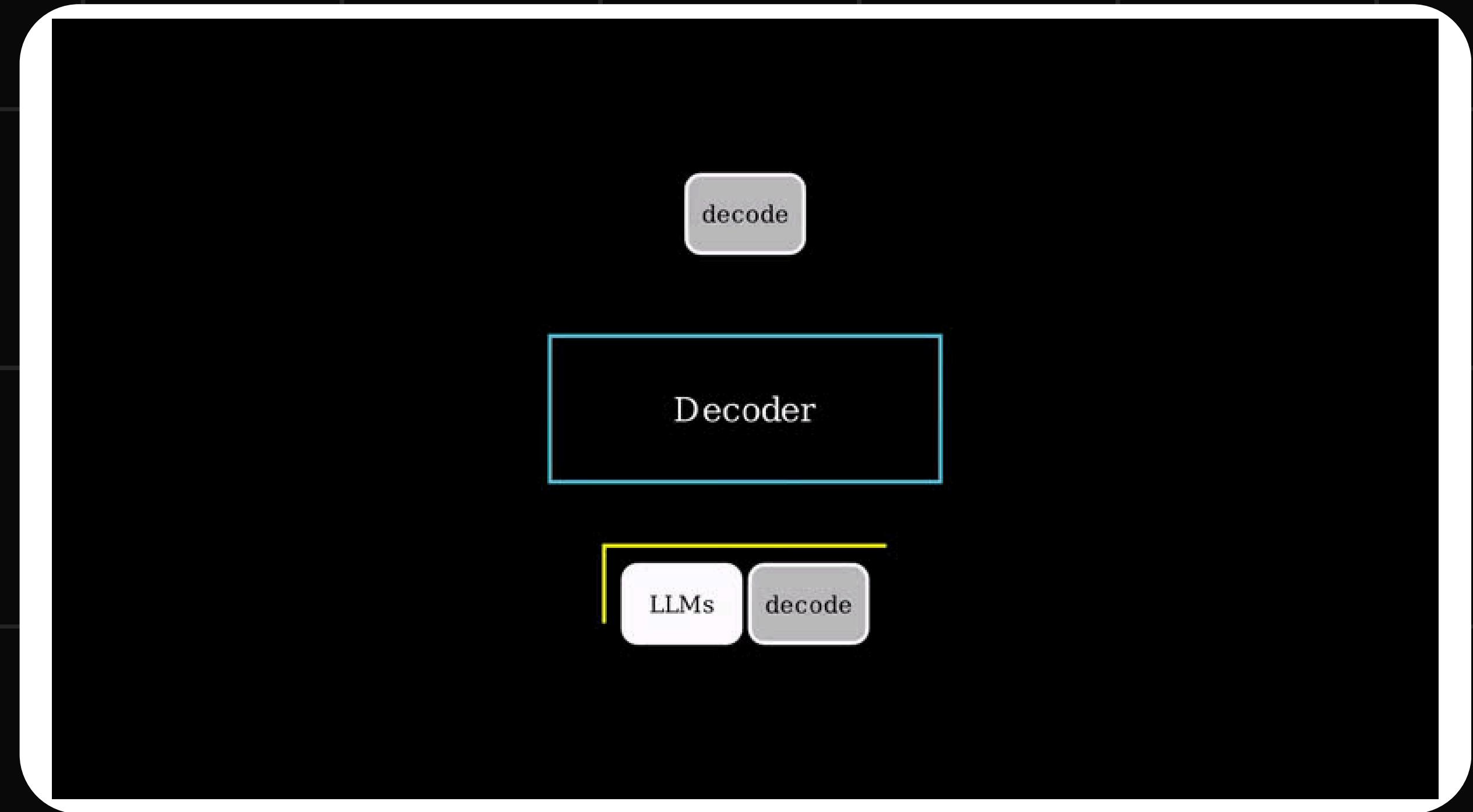
- Tokens são as **unidades fundamentais** de informação processadas por LLMs.
- São frequentemente **unidades de sub-palavras**. Essa abordagem é adotada por **eficiência**.
- um LLM como o Llama 2 pode operar com um **vocabulário de ~32.000 tokens**, significativamente **menos** do que **~600.000 palavras em inglês**.
- Eses tokens de sub-palavra podem ser **combinados para representar o vocabulário completo**.



O que são LLMs ?

PREDIÇÃO DO PROX. TOKEN

- **Autoregressivo**, o que significa que a saída de uma passagem se torna a entrada para a próxima.



O que são LLMs ?

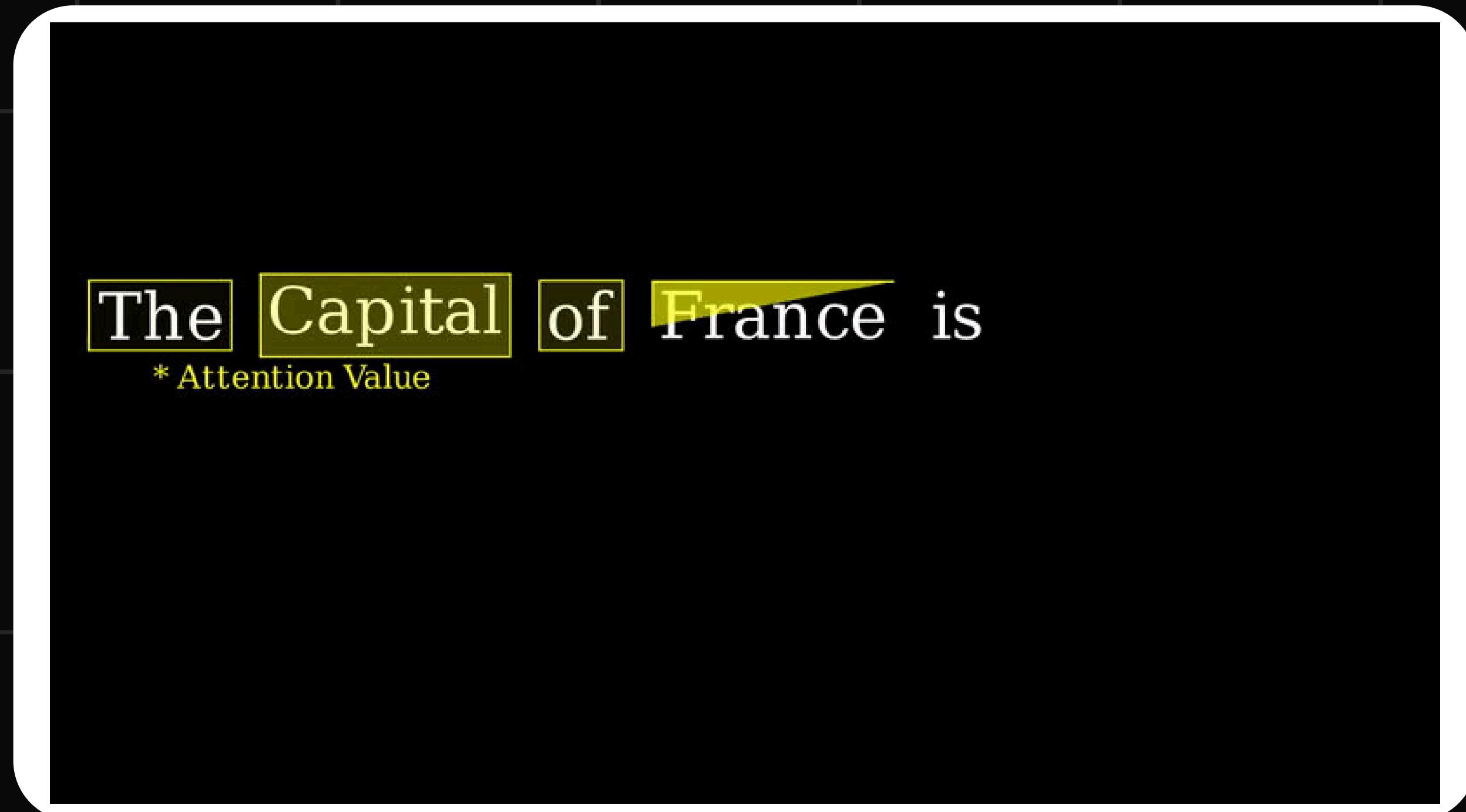
- Uma vez que o texto de entrada é tokenizado, o modelo calcula uma representação da sequência que capta o significado. Essa representação entra no modelo, que gera pontuações que classificam a probabilidade de cada próximo token

We tokenize the prompt

Paris is the city

O que são LLMs ?

- **Atenção.** Ao prever a próxima palavra, nem toda palavra em uma frase é igualmente importante; **EX:** palavras como "França" e "capital" na frase "A capital da França é..." carregam o maior significado.

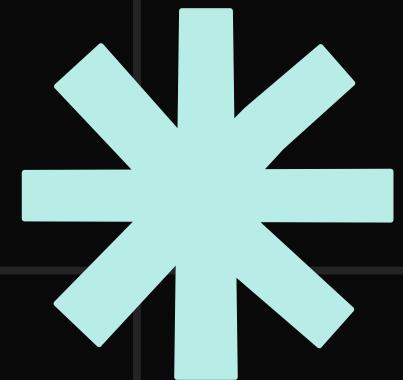


O que são LLMs ?

LIMITAÇÕES IMPORTANTES

- **Alucinações:** Podem gerar informações incorretas com confiança.
- **Falta de compreensão verdadeira:** Eles não possuem uma compreensão verdadeira do mundo e operam puramente com base em padrões estatísticos.
- **Viés:** Podem reproduzir vieses presentes em dados de treinamento.
- **Recursos computacionais e Energéticos:** Exigem recursos computacionais significativos.
- “**AI Enginnering**” envolve **mitigar** estas limitações e construir aplicações em cima destes modelos...

Técnicas de AI Engineering



Técnicas de AI Engineering

PROMPT ENGINEERING

- A Arte de criar **entradas eficazes** para gerar **saídas precisas e úteis**.
- Não existe **UMA forma correta**.
 - Normalmente exige N iterações.
- Muito **subestimado....**
- ***Os 20% que fazem 80% do efeito...***

```
Uma mensagem de desenvolvedor para geração de código

1 # Identity
2
3 You are coding assistant that helps enforce the use of snake case
4 variables in JavaScript code, and writing code that will run in
5 Internet Explorer version 6.
6
7 # Instructions
8
9 * When defining variables, use snake case names (e.g. my_variable)
10 instead of camel case names (e.g. myVariable).
11 * To support old browsers, declare variables using the older
12 "var" keyword.
13 * Do not give responses with Markdown formatting, just return
14 the code as requested.
15
16 # Examples
17
18 <user_query>
19 How do I declare a string variable for a first name?
20 </user_query>
21
22 <assistant_response>
23 var first_name = "Anna";
24 </assistant_response>
```

Prompt Engineering

TÉCNICAS

- **Zero-shot prompting:** O modelo recebe apenas a instrução da tarefa, sem exemplos. É simples e um bom ponto de partida, mas pode ser limitado.
- **One-shot & Few-shot prompting:** Inclui um (one-shot) ou alguns (few-shot) exemplos no prompt para mostrar o padrão esperado.
 - Recomenda-se usar pelo menos 3 a 5 exemplos;
- **System prompting:** Define o propósito geral da tarefa (ex: traduzir, classificar), podendo incluir requisitos de formato (como JSON ou texto em maiúsculas).
- **Contextual prompting:** Adiciona informações específicas e relevantes para a tarefa ou entrada atual. Melhora adaptação e precisão.
- **Role prompting:** Atribui um papel ou persona ao modelo (ex: "Você é um médico"), influenciando estilo, tom e conhecimento aplicado.
- **Combinação de prompting (System + Contextual + Role).**
- **Step-back prompting:** Pede ao modelo para refletir primeiro sobre uma questão mais geral antes de resolver a tarefa específica. Melhora raciocínio e reduz vieses.
- **Chain of Thought (CoT):** Guia o modelo a pensar em etapas antes de responder. Aumenta precisão e interpretabilidade.
- **Self-consistency:** Executa o prompt várias vezes com temperatura alta e seleciona a resposta mais comum.
- **Tree of Thoughts (ToT):** Expande o CoT ao explorar múltiplos caminhos de raciocínio em paralelo.
- **Multimodal prompting:** Usa diferentes tipos de entrada (texto, imagens, áudio, código) para guiar a resposta do modelo.

Prompt Engineering

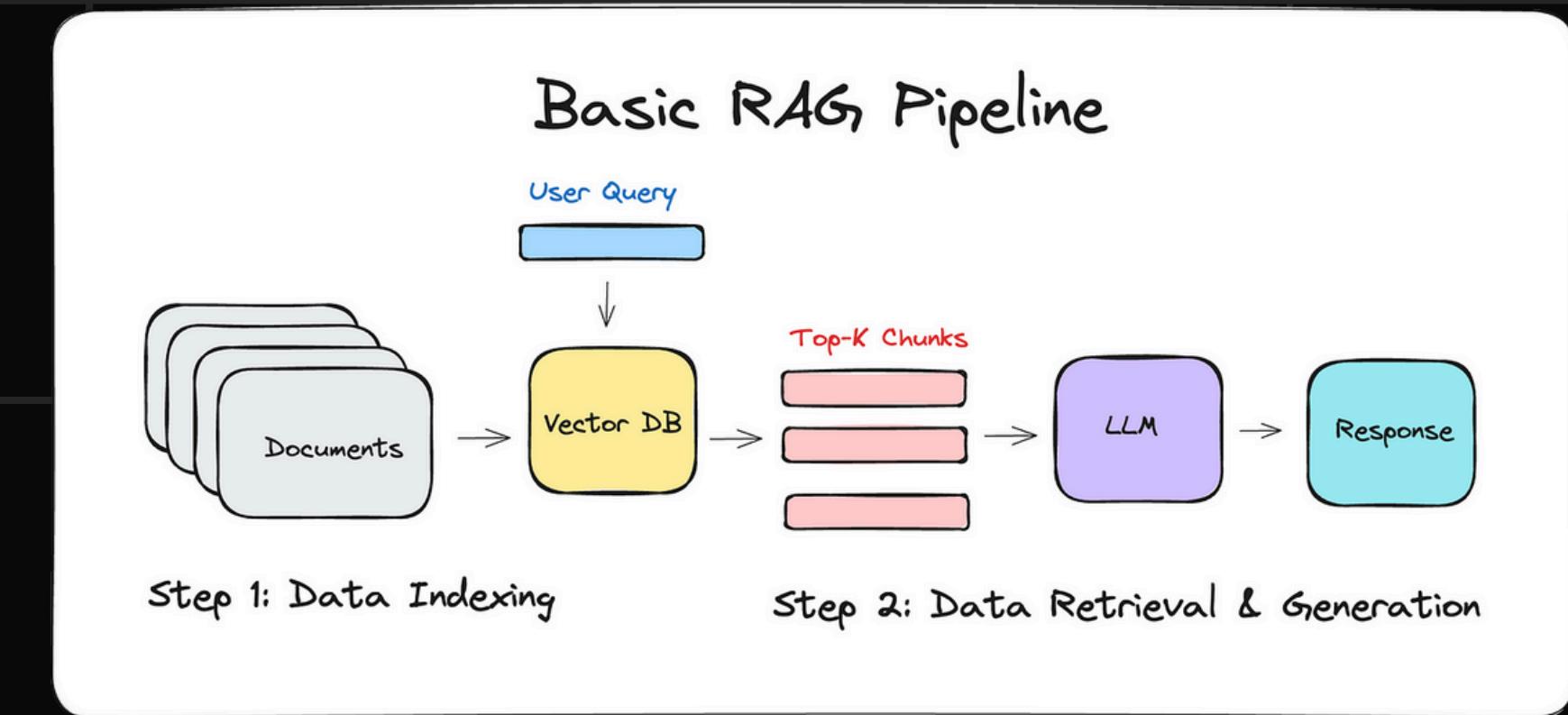
MELHORES PRÁTICAS

- **Forneça exemplos:** Incluir exemplos (one-shot/few-shot) ajuda o modelo a entender melhor a tarefa, melhorando precisão, estilo e tom.
- **Projete com simplicidade:** Use prompts claros e objetivos, com verbos de ação específicos. Evite complexidade desnecessária.
- **Seja específico sobre a saída:** Detalhar o formato e conteúdo esperado melhora a precisão da resposta; instruções vagas tendem a ser ineficazes.
- **Experimente formatos de entrada e estilos de escrita:** Diferentes estilos e configurações afetam os resultados. Teste zero-shot, few-shot, vocabulário, tom etc.
- **Adapte-se às atualizações do modelo:** Ajuste os prompts conforme novas versões dos modelos são lançadas para aproveitar melhor suas capacidades.
- **Experimente formatos de saída (JSON/XML):** Estruturar saídas ajuda a reduzir erros e facilita o uso em sistemas, mas exige cuidado com tamanho e validação.
- **Documente suas tentativas.**

Técnicas de AI Engineering

RAG (RETRIEVAL-AUGMENTED GENERATION)

- LLMs são limitados por serem **estáticos** (conhecimento desatualizado) e carecerem de **conhecimento específico de domínio**, levando a respostas imprecisas ou "alucinações".
- RAG surge como resposta a estes problemas.

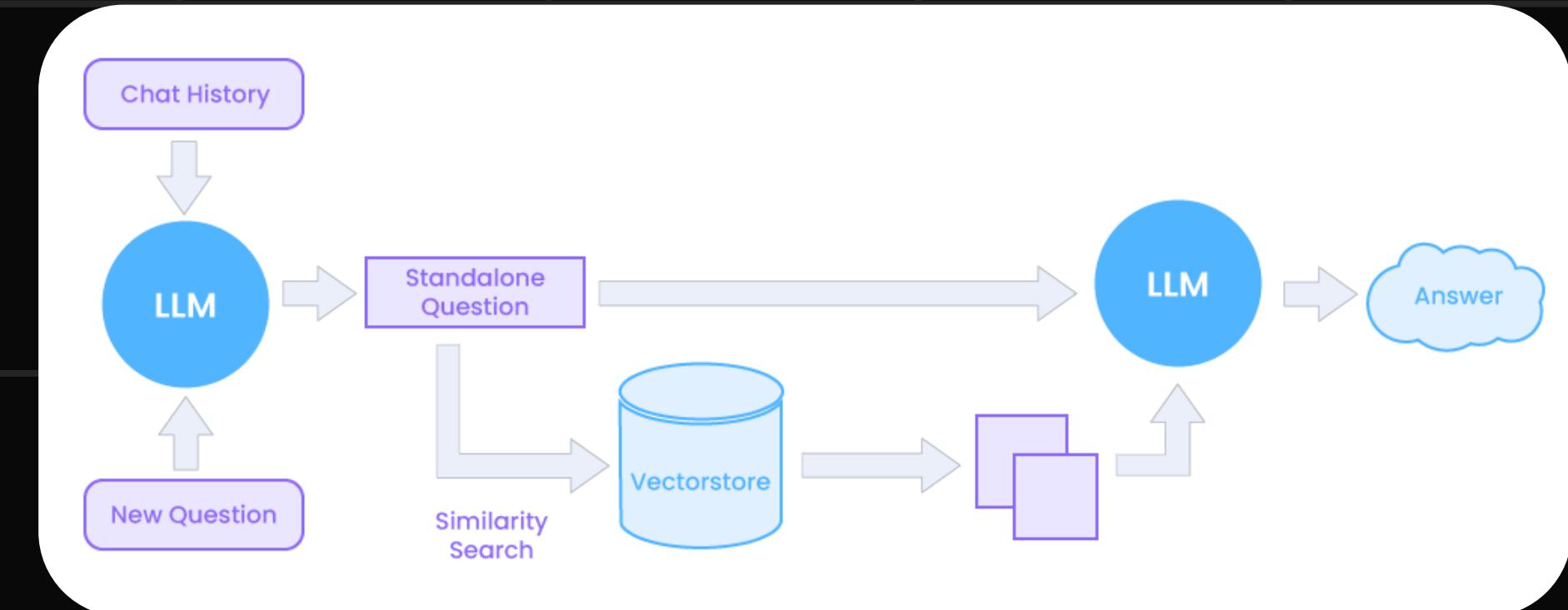


- Traz dados atualizados e específicos de domínio ao modelo.
- Reduz alucinações e melhora a fidelidade da resposta.
- Permite atualização em tempo real dos dados no banco vetorial.

RAG

PASSO A PASSO

- Os **dados** específicos de domínio são **convertidos** em **vetores** usando **modelos de *embedding*** (representam o significado do texto).
- Esse**s vetores** são armazenados em um **banco de dados vetorial**.
- Quando o usuário faz uma **consulta**, ela também é convertida em ***embeddings***.



- A consulta vetorial é usada para realizar uma **busca semântica** no banco de dados (busca por similaridade de significado).
- Os vetores mais relevantes são retornados e enviados ao LLM via janela de contexto.
- O LLM utiliza essas informações para gerar uma resposta mais precisa e contextualizada.

RAG

MELHORIAS

- **Search R1 (RL)**: Um LLM é treinado para buscar e gerar conjuntamente com aprendizado por reforço (**RL**), usando "exact match" como função de recompensa. Foca em **otimizar a query do usuário**.
- **Graph RAG**: Integra grafos de conhecimento no processo. Uma versão baseada em cadeia segue quatro etapas: prever, decompor, buscar e raciocinar.
- **Textual and Visual RAG**: Aplica o RAG a contextos multimodais, especialmente em análise de layout de documentos com entradas textuais e visuais.
- **Agentic RAG**: Adiciona **agentes de IA** ao pipeline de RAG para **aumentar a adaptabilidade e a precisão**, permitindo que LLMs recuperem informações de várias fontes e lidem com fluxos de trabalho mais complexos.

Técnicas de AI Engineering

OUTRAS

- **Fine-tuning:** Treinamento adicional que adapta um LLM fundacional para necessidades específicas de uma aplicação.
- **Distillation:** é um processo para criar uma versão menor de um LLM, com menos parâmetros, que gera previsões muito mais rápidas e requer menos recursos computacionais e ambientais do que o LLM completo. Uma das formas é colocar o modelo maior para ensinar o menor.
- **Parameter-Efficient Tuning (PEFT):** Ajusta apenas um subconjunto de parâmetros do modelo, em vez de todos eles, tornando o processo computacionalmente menos caro do que o fine-tuning padrão, mesmo com um número relativamente pequeno de exemplos de treinamento.

Técnicas de AI Engineering

QUESTÕES DE SEGURANÇA

- **Prompt Injection** (Injeção de Prompt): Atacantes inserem instruções maliciosas nos prompts para manipular o comportamento do LLM, buscando divulgar informações, gerar conteúdo indesejado ou executar ações não autorizadas.
- **Insecure Output Handling** (Tratamento Inseguro da Saída): A saída do LLM, se não for validada e sanitizada, pode levar a vulnerabilidades tradicionais de segurança (ex: XSS, SQL Injection) ao ser usada em outras partes da aplicação.
- **Divulgação de Informações Sensíveis**: LLMs podem inadvertidamente expor dados sensíveis.
- **Excessive Agency** (Agência Excessiva): Conceder permissões e acessos excessivos a um LLM pode resultar em ações não intencionais e prejudiciais, caso o modelo seja manipulado.

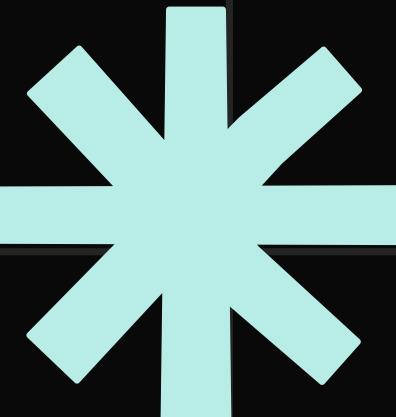
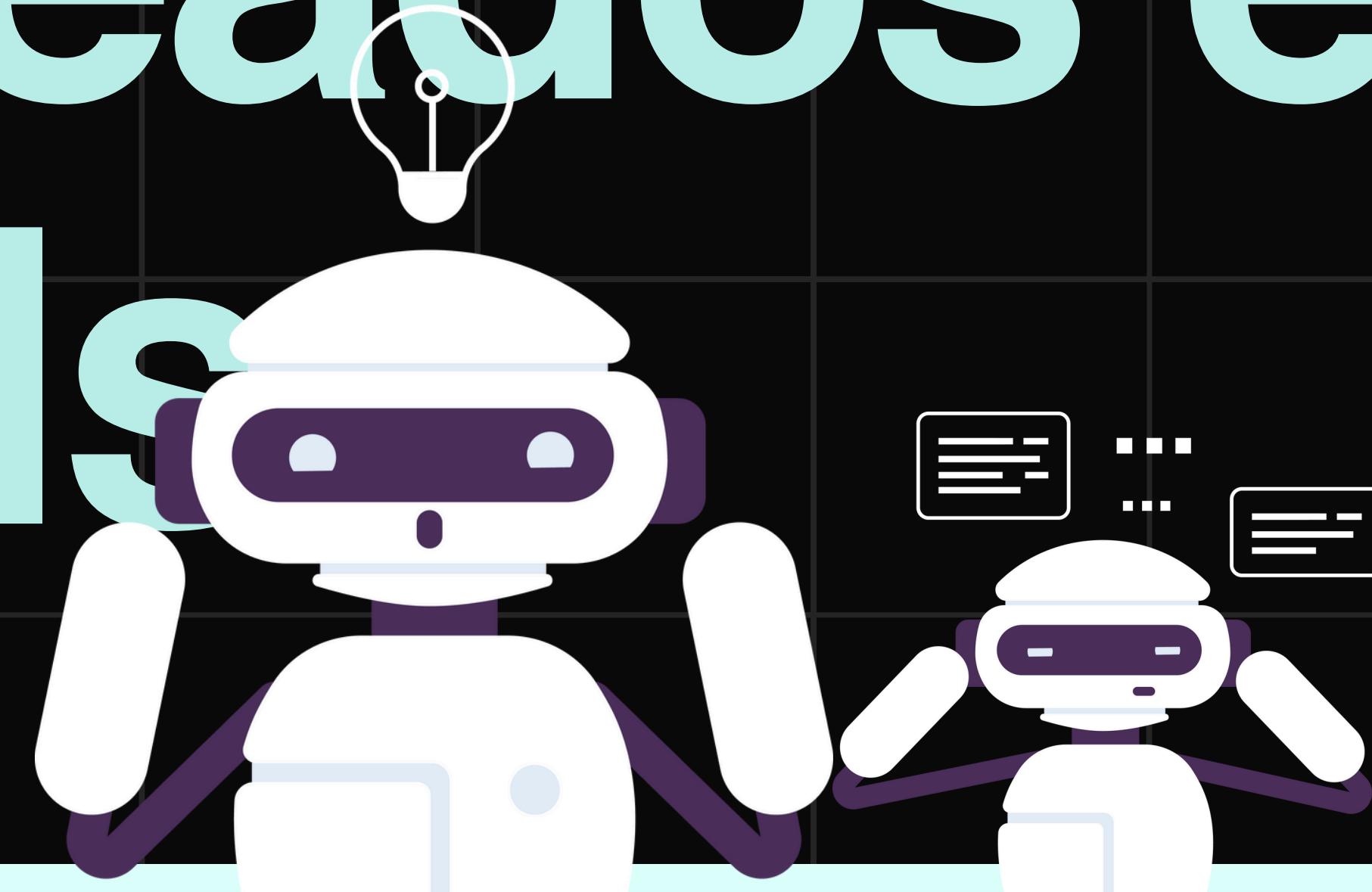
Técnicas de AI Engineering

SEGURANÇA - MELHORES PRÁTICAS

- **Validação e Sanitização de Entradas e Saídas:** Implementar filtros rigorosos para prompts e garantir que todas as saídas do LLM sejam limpas antes de serem usadas.
- **Princípio do Privilégio Mínimo:** Conceder aos LLMs e seus componentes apenas as permissões e acessos estritamente necessários para suas funções.
- **Monitoramento e Observabilidade:** Acompanhar de perto o comportamento do LLM em produção para detectar atividades anômalas ou maliciosas.
- **Guardrails:** Desenvolver mecanismos para filtrar e bloquear a geração de conteúdo tóxico, enviesado ou prejudicial.
- **Transparência e Explicabilidade (XAI):** Buscar entender como o LLM toma decisões para identificar comportamentos inesperados e aumentar a confiança.

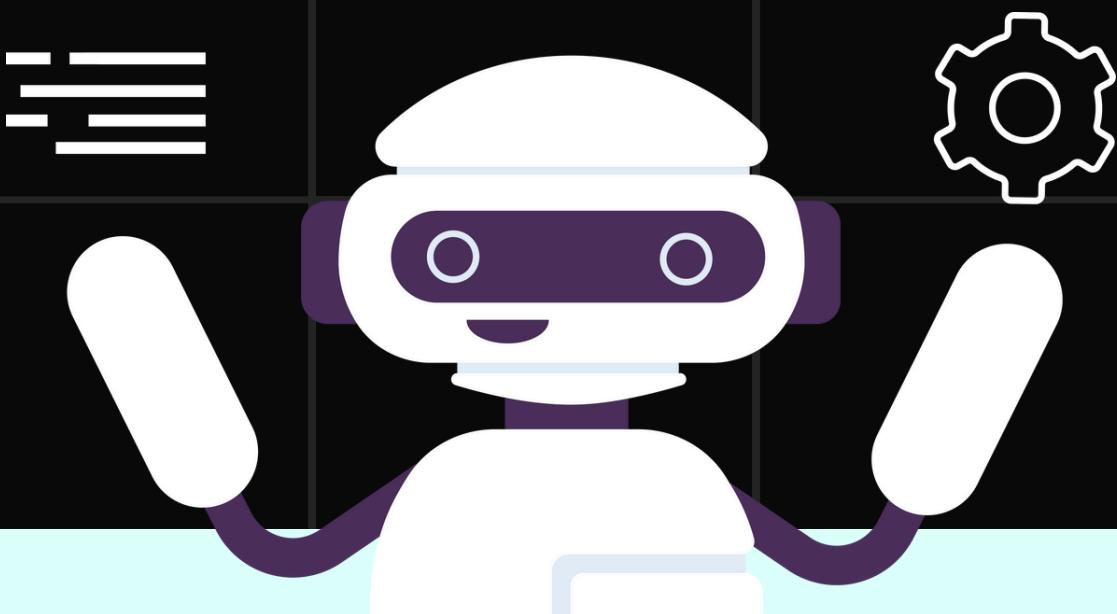
Agentes baseados em

LLMs



Agentes baseados em LLMs

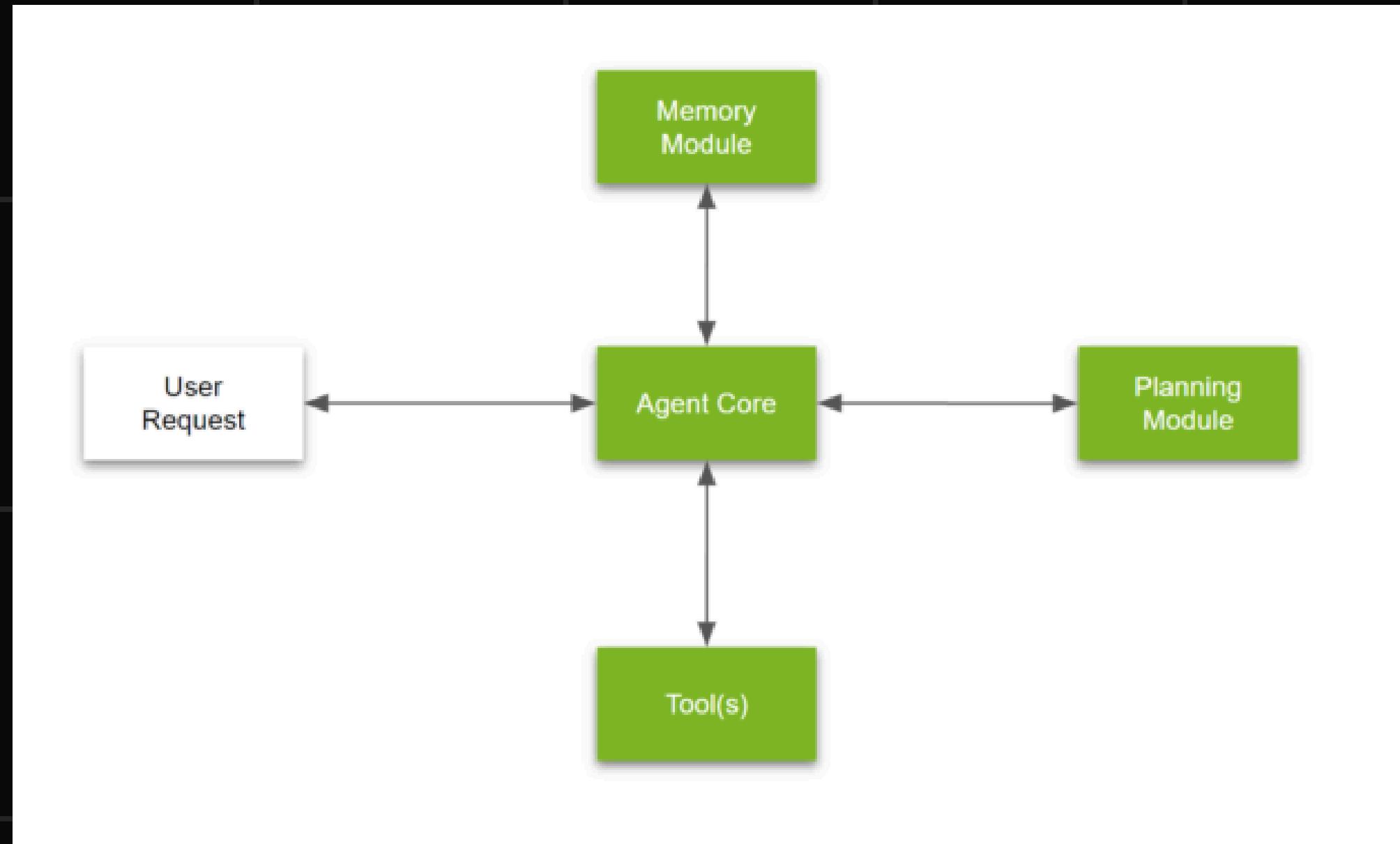
- **Definição:** “é um sistema que aproveita um LLM para interagir com seu ambiente a fim de alcançar um objetivo definido pelo usuário. Ele combina raciocínio, planejamento e a execução de Ações (frequentemente por meio de *Tools* externas) para cumprir tarefas.”
- **Duas partes principais**
 - O “Cérebro” (LLM): lida com o raciocínio e o planejamento, decidindo quais Ações tomar com base na situação.
 - O Corpo (Capacidades e *Tools*): representa tudo o que o Agente está equipado para fazer.



Agentes baseados em LLMs

- **Agent Core:** Atua como o módulo chave para a tomada de decisões do agente.
- **Memory Module:** Funciona como o repositório interno das ações e interações do agente. Possui memória de curto e longo prazo.
- **Tools:** São workflows executáveis ou APIs especializadas acessíveis ao agente. EX: Pipeline RAG.
- **Planning Module (LLM):** Responsável por resolver tarefas complexas que exigem múltiplas etapas ou raciocínio refinado.

Arquitetura Genérica de um Agente de LLM



Agentes baseados em LLMs

TOOLS

- **O Que São:** Funções que **dão aos Agentes capacidades extras**, como realizar cálculos ou acessar dados externos.
- **Como Definir:** Fornecendo uma **descrição textual clara, entradas, saídas...**
- **Por Que São Essenciais:** Elas permitem que os Agentes superem as limitações, **interajam com seu ambiente, lidem com tarefas em tempo real e executem ações especializadas.**

Ferramenta	Descrição
Pesquisa na Web	Permite que o agente obtenha informações atualizadas da web.
Geração de imagem	Cria imagens com base em descrições de texto.
Recuperação	Recupera informações de uma fonte externa.
Interface da API	Interage com uma API externa (GitHub, YouTube, Spotify).

```
def calculator(a: int, b: int) -> int:  
    """Multiply two integers."""  
    return a * b
```

Agentes baseados em LLMs

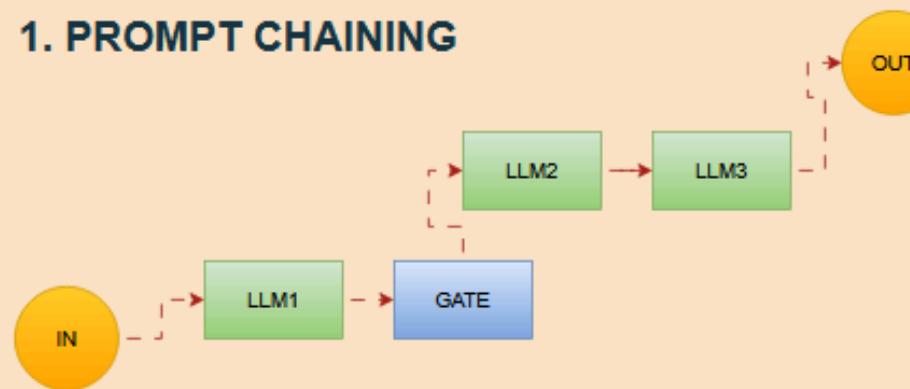
MULTI-AGENTS

- **O que são?** Consiste em vários agentes que trabalham para executar tarefas complexas em nome de um usuário ou de outro sistema.
- **Quais problemas resolvem?** Resolvem problemas como um agente **ter muitas ferramentas e tomar decisões ruins**, o **contexto se tornar muito complexo para um único agente rastrear**, ou a **necessidade de múltiplas áreas de especialização**.
- **Como se diferem de sistemas com único agente?** Sistemas multiagentes envolvem todos os agentes do ambiente para modelar os objetivos, a memória e o plano de ação uns dos outros. Tendem a superar sistemas de agente único devido ao maior conjunto de recursos compartilhados, otimização e automação.
- **Exemplos de domínios aplicáveis:** sistemas de transporte, várias tarefas na área da saúde, como previsão e prevenção de doenças, gerenciamento da cadeia de suprimentos, e fortalecimento de sistemas de defesa.

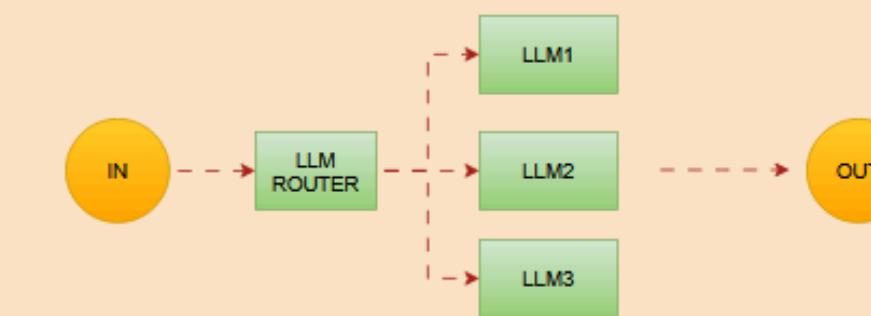
Multiagent/Workflow PATTERNS

5 Design Patterns in Agentic AI Workflow

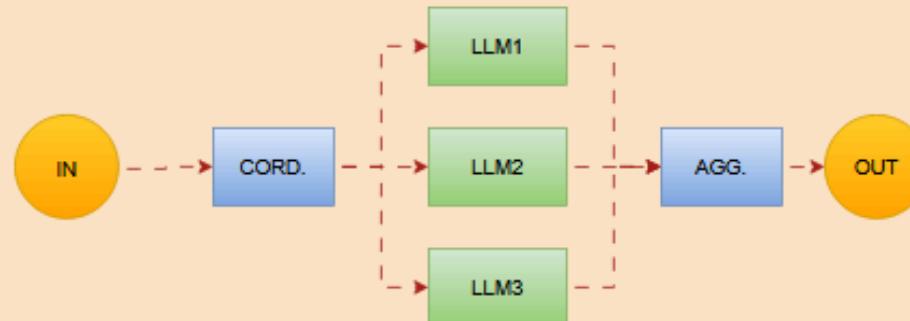
1. PROMPT CHAINING



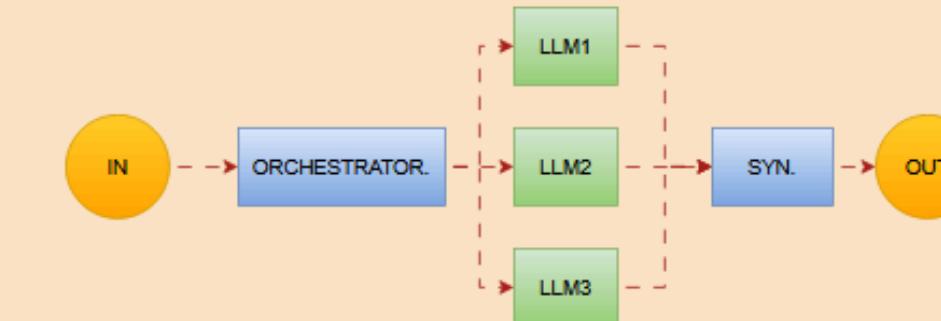
2. ROUTING



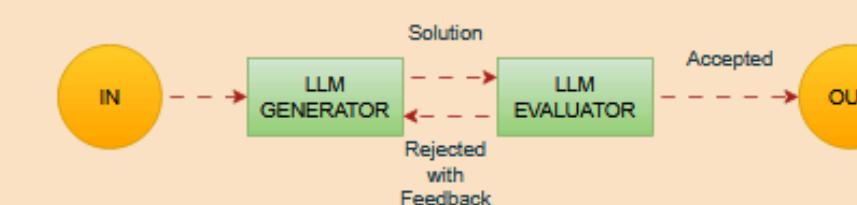
3. PARALLELIZATION



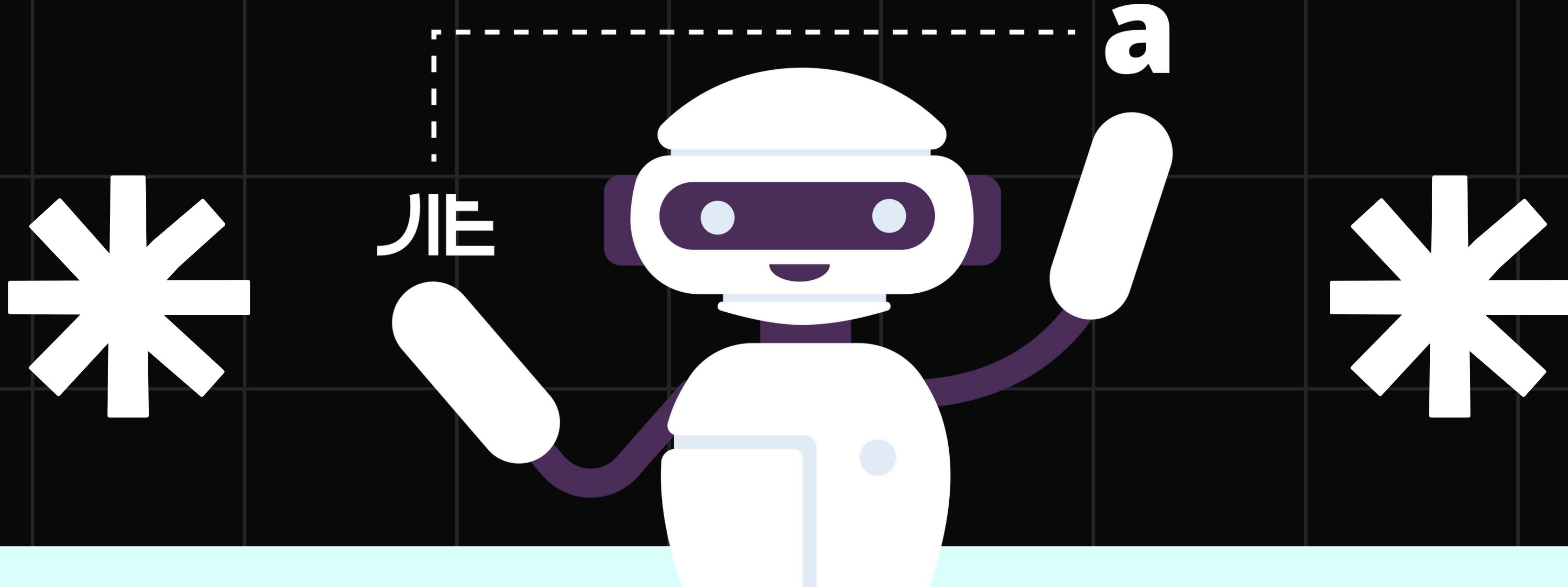
4. ORCHESTRATOR-WORKER



5. EVALUATOR-OPTIMIZER

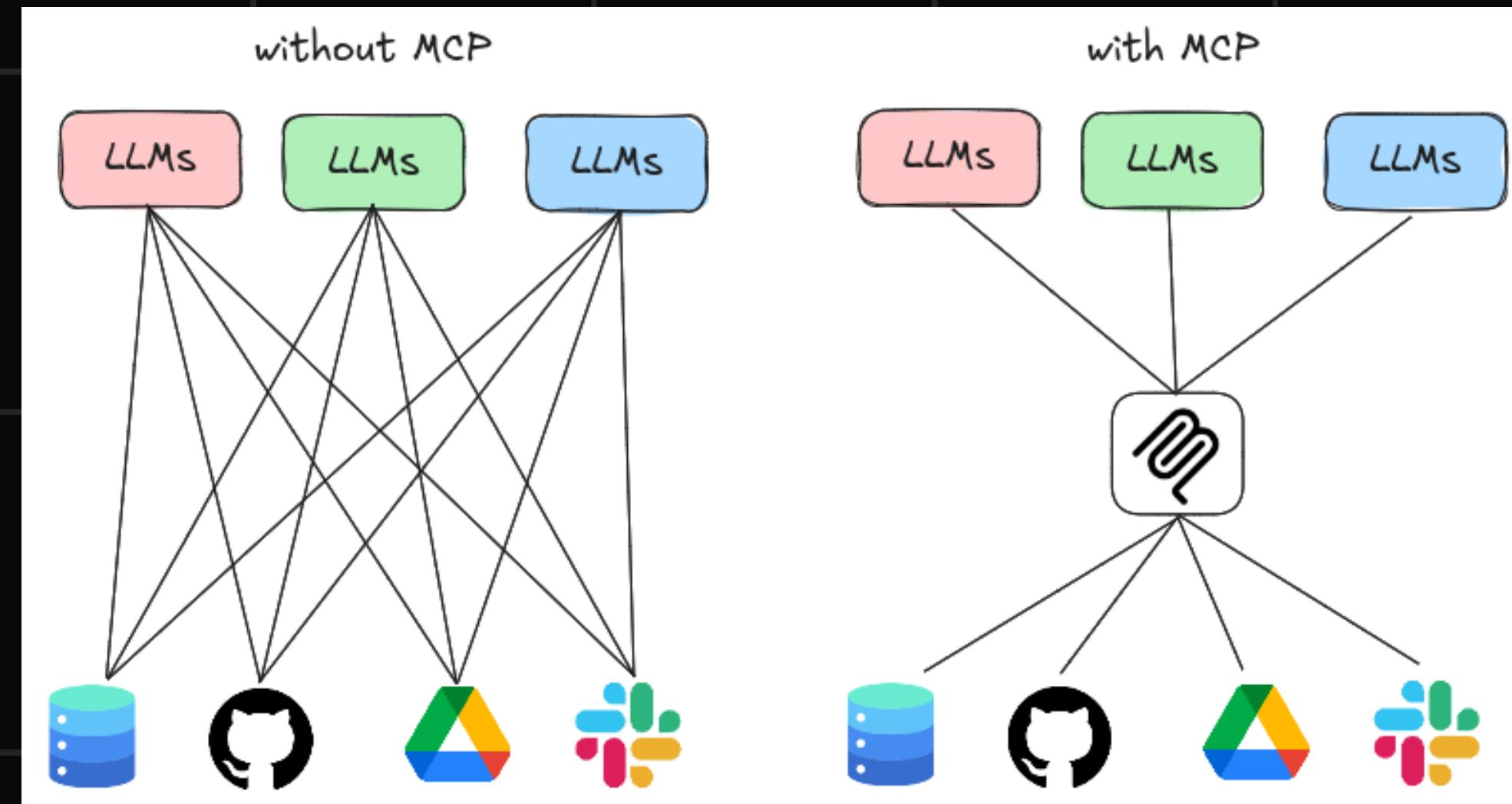


Model Context Protocol (MCP)

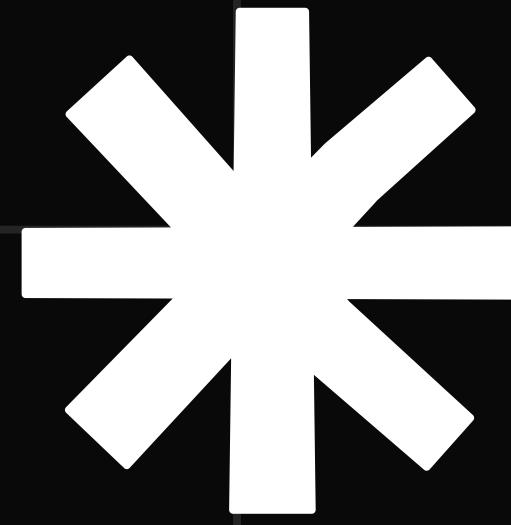


MCP

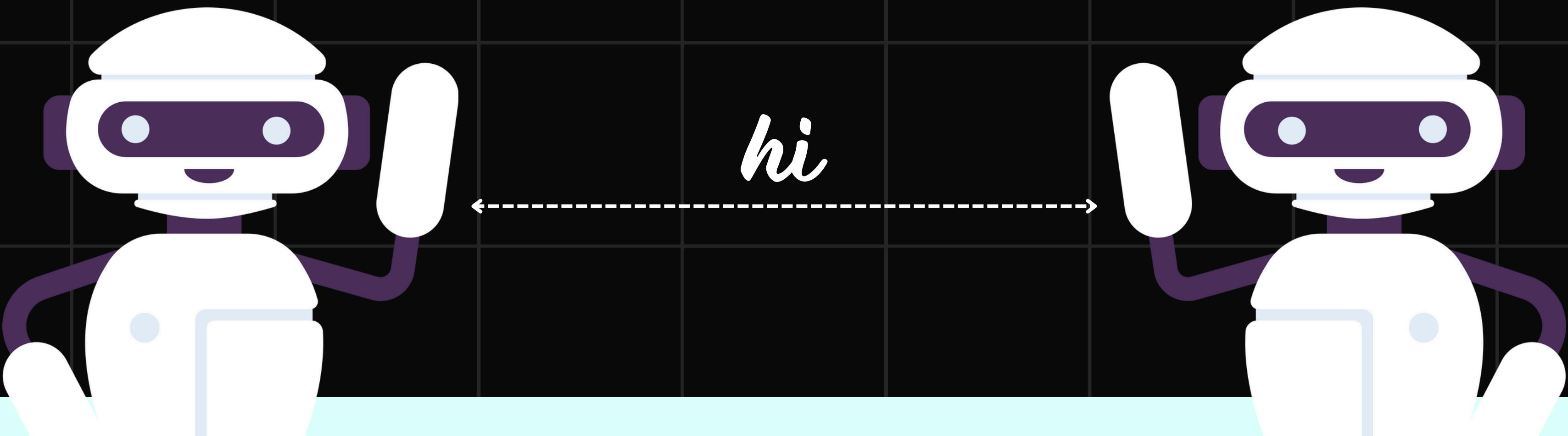
- Padrão universal projetado para aprimorar a interação entre **Agentes de LLM e Tools**. Inspira-se no HTTP.
- Propósito Principal: Ajudar os modelos de IA a gerar respostas mais **precisas e contextualmente relevantes**.
- Arquitetura Flexível: Os desenvolvedores podem expor seus dados via servidores MCP ou criar aplicações (clientes MCP) que se conectam a esses servidores.
- Manutenção do Contexto: Permite que os sistemas de IA mantenham o contexto mesmo ao interagir com diferentes ferramentas e conjuntos de dados.



- Acesso Simplificado aos Dados: Facilita e torna mais confiável o acesso dos sistemas de IA aos dados necessários.

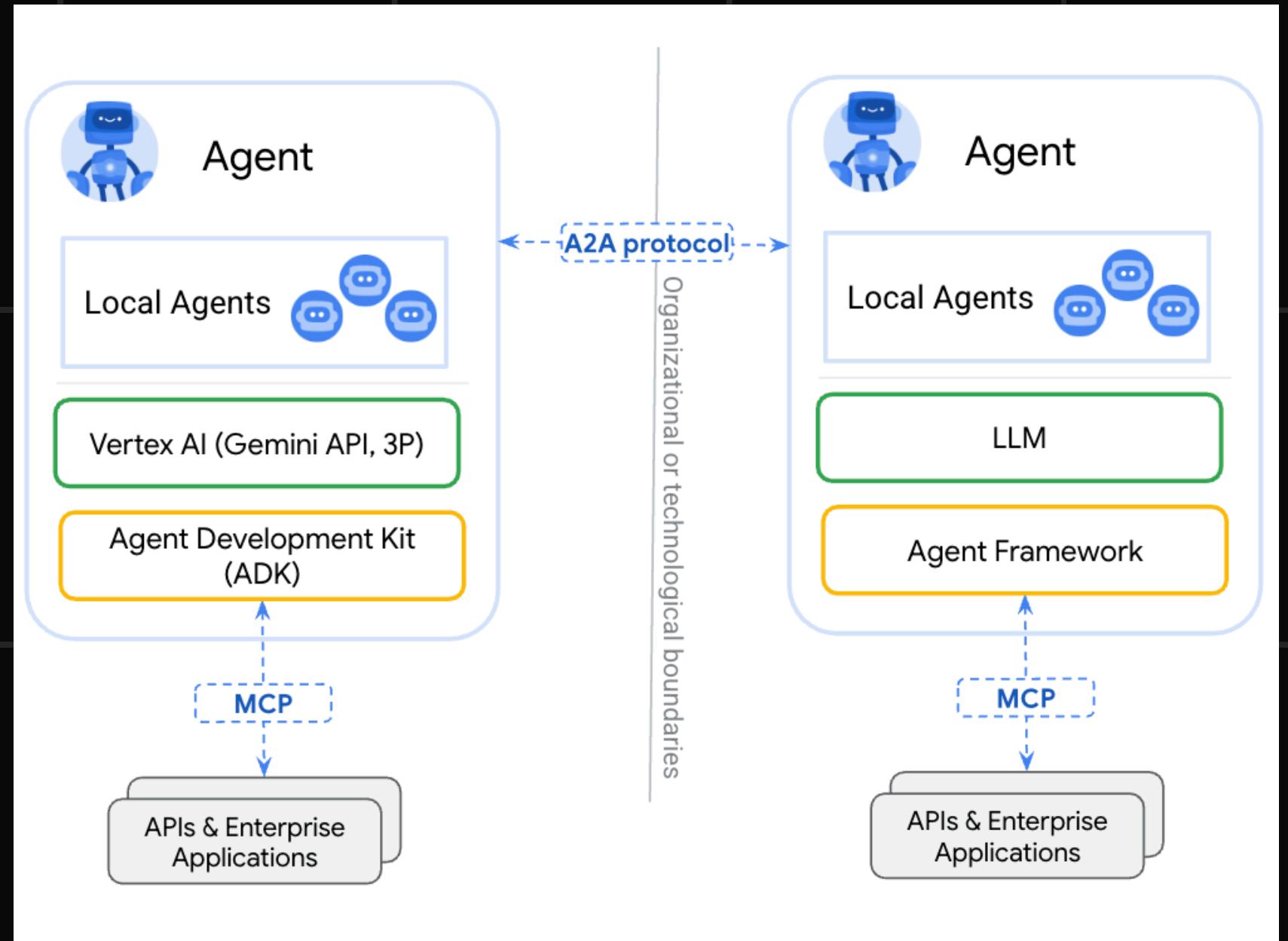


Agent2Agent Protocol

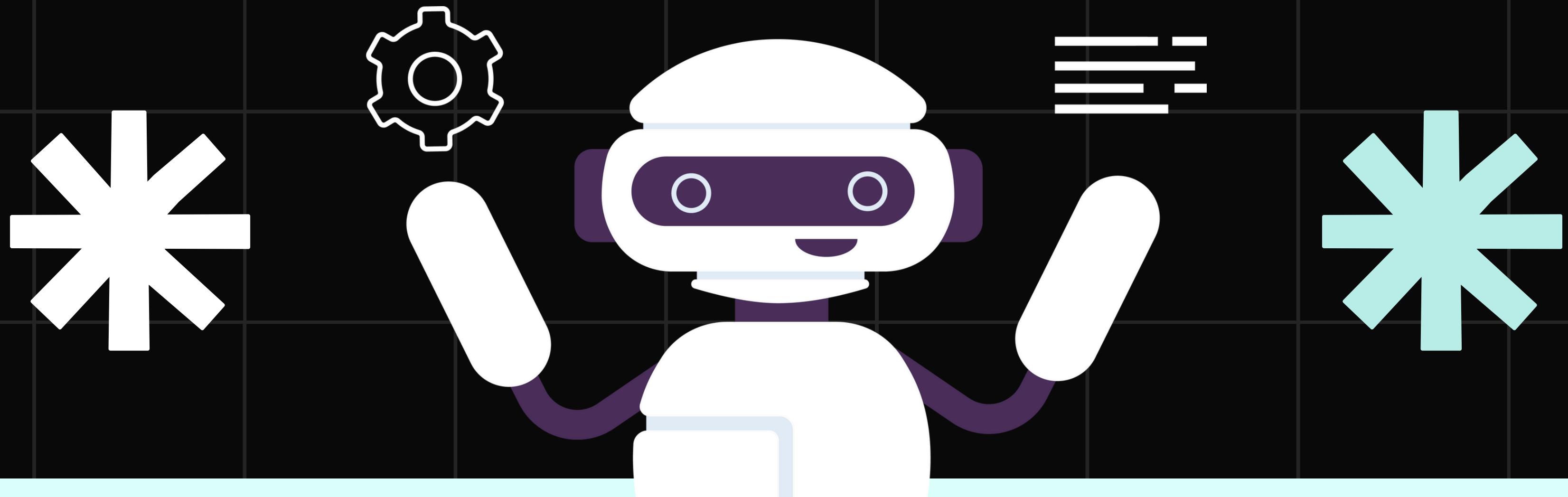


A2A

- Busca fornecer uma **camada de comunicação padronizada**, permitindo que agentes construídos em plataformas diferentes se comuniquem.
- Agentes podem enviar mensagens uns aos outros para **comunicar contexto, respostas, artefatos ou instruções do usuário**.
- É construído sobre **padrões existentes e populares**, incluindo **HTTP, SSE, JSON-RPC**, tornando-o mais fácil de integrar.
- É **agnóstico em termos de mídia**.



Frameworks para criação de Agentes

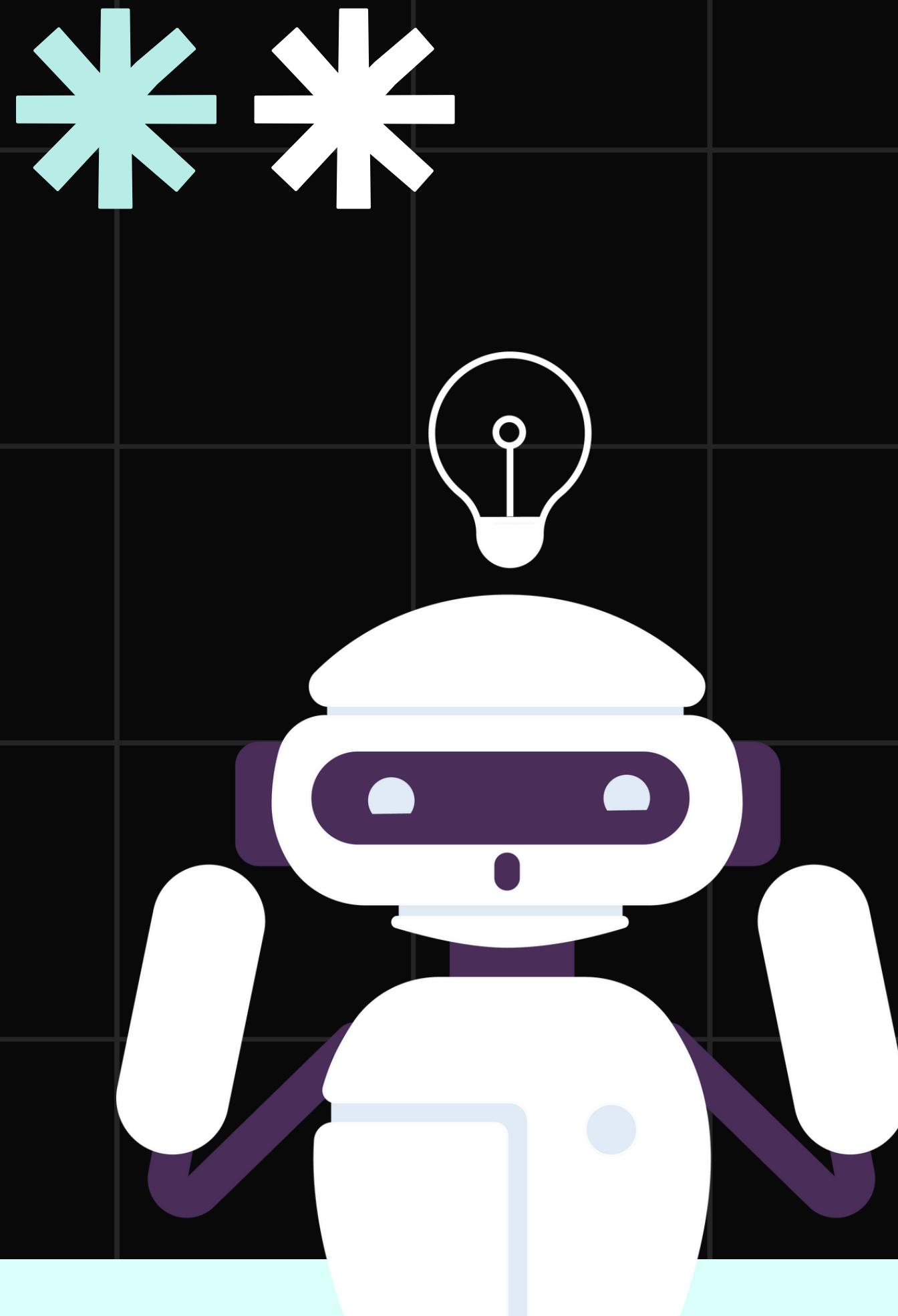


Frameworks

- **Bom para prototipação:** smolagents (HuggingFace), CrewAI.
- **Market proof:** LangChain, LangGraph, Microsoft AutoGen,...
- **Bons em coisas específicas:** LlamalIndex.
- **Bom no geral:** PydanticAI.
- **Promissores:** Google ADK



Casos de USO

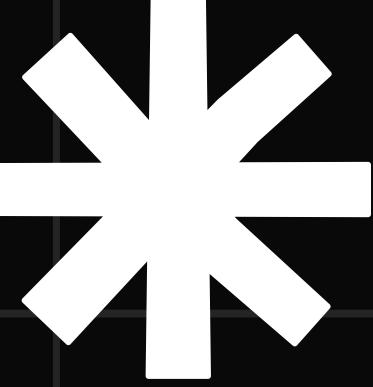


Casos de Uso

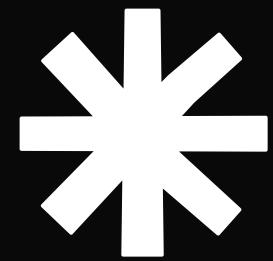
AGENTES

- <https://cloud.google.com/transform/101-real-world-generative-ai-use-cases-from-industry-leaders>

Links úteis



- Google Prompt Guide - https://drive.google.com/file/d/1AbaBYbEa_EbPelsT40-vj64L-2IwUJHy/view?usp=drive_link
- <https://www.pinecone.io/learn/retrieval-augmented-generation/>
- <https://cdn.openai.com/business-guides-and-resources/a-practical-guide-to-building-agents.pdf>
- <https://cdn.openai.com/business-guides-and-resources/identifying-and-scaling-ai-use-cases.pdf>
- <https://huggingface.co/learn/agents-course>
- <https://huggingface.co/learn/cookbook>
- <https://google.github.io/adk-docs/>
- <https://www.anthropic.com/engineering/building-effective-agents>
- <https://docs.anthropic.com/en/docs/agents-and-tools/mcp>



QR CODE MATERIAIS



linkedin.com/in/natthan-elias



natthandosantos@gmail.com

