UNIVERSIDADE FEDERAL DE GOIÁS / INSTITUTO DE INFORMÁTICA

Sistemas Multiagentes Baseados em LLMs

Uma Abordagem orientada à Tomada de Decisão em Mercados Financeiros

Alberto Lucas B. A. Teixeira

UFG
UNIVERSIDADE FEDERAL DE GOIÁS

UNIVERSIDADE FEDERAL DE GOIÁS (UFG)
INSTITUTO DE INFORMÁTICA (INF)

ALBERTO LUCAS BORGES DE ALMEIDA TEIXEIRA

Sistemas Multiagentes Baseados em LLMs

Uma Abordagem orientada à Tomada de Decisão em Mercados Financeiros

Goiânia
2025

UNIVERSIDADE FEDERAL DE GOIÁS
INSTITUTO DE INFORMÁTICA

TERMO DE CIÊNCIA E DE AUTORIZAÇÃO PARA DISPONIBILIZAR VERSÕES ELETRÔNICAS DE TRABALHO DE CONCLUSÃO DE CURSO DE GRADUAÇÃO NO REPOSITÓRIO INSTITUCIONAL DA UFG

Na qualidade de titular dos direitos de autor, autorizo a Universidade Federal de Goiás (UFG) a disponibilizar, gratuitamente, por meio do Repositório Institucional (RI/UFG), regulamentado pela Resolução CEPEC no 1240/2014, sem ressarcimento dos direitos autorais, de acordo com a Lei no 9.610/98, o documento conforme permissões assinaladas abaixo, para fins de leitura, impressão e/ou download, a título de divulgação da produção científica brasileira, a partir desta data.

O conteúdo dos Trabalhos de Conclusão dos Cursos de Graduação disponibilizado no RI/UFG é de responsabilidade exclusiva dos autores. Ao encaminhar(em) o produto final, o(s) autor(a)(es)(as) e o(a) orientador(a) firmam o compromisso de que o trabalho não contém nenhuma violação de quaisquer direitos autorais ou outro direito de terceiros.

1. Identificação do Trabalho de Conclusão de Curso de Graduação (TCCG)

Nome(s) completo(s) do(a)(s) autor(a)(es)(as): ALBERTO LUCAS BORGES DE ALMEIDA TEIXEIRA

Título do trabalho: Sistemas Multiagentes Baseados em LLMs

Uma Abordagem orientada à Tomada de Decisão em Mercados Financeiros

2. Informações de acesso ao documento (este campo deve ser preenchido pelo orientador) Concorda com a liberação total do documento [X] SIM [ ] NÃO[1]

[1] Neste caso o documento será embargado por até um ano a partir da data de defesa. Após esse período, a possível disponibilização ocorrerá apenas mediante: a) consulta ao(à)(s) autor(a)(es)(as) e ao(à) orientador(a); b) novo Termo de Ciência e de Autorização (TECA) assinado e inserido no arquivo do TCCG. O documento não será disponibilizado durante o período de embargo.

Casos de embargo:

- Solicitação de registro de patente;

- Submissão de artigo em revista científica;

- Publicação como capítulo de livro.

Obs.: Este termo deve ser assinado no SEI pelo orientador e pelo autor.

Documento assinado eletronicamente por **Alberto Lucas Borges De Almeida Teixeira** , **Discente**, em 04/02/2026, às 16:15, conforme horário oficial de Brasília, com fundamento no § 3º do art. 4º do Decreto nº 10.543, de 13 de novembro de 2020 .

Documento assinado eletronicamente por **Fernando Marques Federson**, **Professor do Magistério Superior**, em 13/03/2026, às 11:14, conforme horário oficial de Brasília, com fundamento no § 3º do art. 4º do Decreto nº 10.543, de 13 de novembro de 2020 .

A autenticidade deste documento pode ser conferida no site https://sei.ufg.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0, informando o código verificador **5956200** e o código CRC **03B6DA50**.

Referência: Processo nº 23070.005470/2026-89 SEI nº 5956200

ALBERTO LUCAS BORGES DE ALMEIDA TEIXEIRA

Sistemas Multiagentes Baseados em LLMs

Uma Abordagem orientada à Tomada de Decisão em Mercados Financeiros

Relatório final de Trabalho de Conclusão de Curso, apresentado à Universidade Federal de Goiás, como parte das exigências para a obtenção do título de Bacharel em Inteligência Artificial.
Orientador: Prof. Dr. Fernando Marques Federson

Goiânia

2025

Ficha de identificação da obra elaborada pelo autor, através do Programa de Geração Automática do Sistema de Bibliotecas da UFG.

TEIXEIRA, ALBERTO LUCAS BORGES DE ALMEIDA
Sistemas Multiagentes Baseados em LLMs [manuscrito]: Uma Abordagem orientada à Tomada de Decisão em Mercados Financeiros / ALBERTO LUCAS BORGES DE ALMEIDA TEIXEIRA. - 2025.
109 f.: 2025

Orientador: Prof. Dr. Fernando Marques Federson
Trabalho de Conclusão de Curso (Graduação) - Universidade Federal de Goiás, Instituto de Informática (INF), Inteligência Artificial, Goiânia, 2025.

1. Inteligência Artificial. 2. Agentes Inteligentes. 3. Sistemas Multiagentes.

I. Federson, Fernando Marques , orient. II. Título.

CDU 004

ALBERTO LUCAS BORGES DE ALMEIDA TEIXEIRA

Sistemas Multiagentes Baseados em LLMs

Uma Abordagem orientada à Tomada de Decisão em Mercados Financeiros

Relatório final de Trabalho de Conclusão de Curso, apresentado à Universidade Federal de Goiás, como parte das exigências para a obtenção do título de Bacharel em Inteligência Artificial.

Data da Aprovação: 09 de dezembro de 2025.

Prof. Dr. Fernando Marques Federson
Orientador (INF-UFG)

Prof. Dr. Aldo André Díaz Salazar
Coordenador de TCC do BIA (INF-UFG)

Prof. Dr. Anderson da Silva Soares
Coordenador do BIA (INF-UFG)

Prof. Dr. Iwens Gervasio Sene Junior
(INF-UFG)

ALBERTO LUCAS BORGES DE ALMEIDA TEIXEIRA

Sistemas Multiagentes Baseados em LLMs

Uma Abordagem orientada à Tomada de Decisão em Mercados Financeiros

RESUMO

Este Relatório de Conclusão de Curso tem como objetivo reunir os resultados da minha jornada para me tornar um especialista em **Sistemas Multiagentes**. Uma ilustração e sua narrativa descrevem os períodos de trabalho. Os Apêndices contêm os Termos de Aceite de Entrega e os resultados obtidos durante cada período de trabalho.

Palavras-chave: Inteligência artificial; Agentes inteligentes; Sistemas multiagentes.

ABSTRACT

This Course Completion Report aims to bring together the results of my journey to become an expert in **Multi-agent systems**. An illustration and its narrative describe the work periods. The Appendices contain the Delivery Acceptance Terms and the results obtained during each work period.

Keywords: Artificial intelligence; Intelligent agents; Multi-agent systems.

Goiânia

2025

Minha Jornada

Especialista em: Sistemas Multiagentes

Alberto Lucas Borges de Almeida Teixeira

MINHA JORNADA

Nome: **Alberto Lucas Borges de Almeida Teixeira**

Especialidade: **Sistemas Multiagentes**

Objetivo deste documento

Durante o processo da disciplina Residência em IA[1], foram gerados diversos resultados na construção da minha especialização. A cada semana, um conjunto de resultados foi formalizado por um Termo de Aceite de Entrega e avaliado por uma banca, considerando o planejado e o realizado para o período. Este documento tem como objetivo descrever esses resultados obtidos, fazendo referência aos Termos de Aceite de Entrega e seus documentos associados.

Minha Jornada

Minha jornada teve início com o interesse em três possíveis temas: Agentes Inteligentes, Processamento de Linguagem Natural e Sistemas de Aprendizagem Multimodal. Nas **Semanas 1 e 2**, me dediquei a analisar os tópicos dessas áreas nas conferências ICAI'25, ICDATA'25 e ACC'25, processo que me permitiu comparar essas linhas de pesquisa e, consequentemente, escolher Agentes Inteligentes como o foco de aprofundamento na Residência. Após essa definição, iniciei os estudos fundamentais com a leitura dos capítulos 1 e 2 do livro "Inteligência Artificial: Uma Abordagem Moderna" e do artigo basilar "Intelligent Agents: Theory and Practice", além de elaborar um método próprio para seleção de materiais inspirado em revisões de literatura, cujos resultados e anotações constam no **Apêndice 1**.

Durante o período das **Semanas 3, 4 e 5**, restringi o escopo para agentes baseados em LLMs, migrando de uma visão mais geral de agentes para um recorte focado em LLM-based agents. **Na Semana 3**, estudei o survey "A Survey on Large Language Model

[1] Dez Semanas, entre setembro de 2025 e dezembro de 2025.

based Autonomous Agents", que apresenta um framework unificado para construção, aplicação e avaliação de agentes, discute capacidades, gargalos e tendências futuras, e serviu como base para mapear as principais subáreas da área. Nas **Semanas 4 e 5**, utilizei o método de seleção de artigos desenvolvido anteriormente para organizar a literatura em eixos temáticos, estruturando os estudos em arquitetura, aplicações e sistemas multiagentes, aliados a temas de otimização, segurança e avaliação, migrando os materiais para uma planilha que facilitou o acompanhamento sistemático das leituras. Nesse período, estudei surveys importantes sobre otimização de LLM-based agents, planejamento e sistemas multiagentes baseados em LLMs, refinando tanto minha compreensão conceitual quanto o direcionamento da pesquisa. Os resumos, mapas de subáreas e registros das leituras realizadas ao longo das **Semanas 3, 4 e 5** podem ser encontrados no **Apêndice 2**.

As **Semanas 6 e 7** marcaram uma transição importante da teoria para uma aplicação mais concreta no mercado financeiro. Na **Semana 6**, direcionei meus esforços para aplicações práticas, com foco especial no estudo do trabalho "TradingAgents: Multi-Agents LLM Financial Trading Framework". Este trabalho propõe um sistema multiagente inspirado na dinâmica de uma firma de trading, com agentes especializados na geração de relatórios de análise fundamentalista, técnica, de notícias e sentimento, documentos que servem de base para as camadas hierárquicas de tomada de decisão e gestão de risco. Esse artigo foi decisivo para a definição do tema da Residência: Análise Preditiva de Ações com Arquitetura Multiagente, motivando também o estudo panorâmico de frameworks para orquestração de agentes realizada ainda nesse período. Na **Semana 7**, aprofundei-me especificamente no LangGraph, escolha justificada por ser o framework utilizado na construção do TradingAgents, e avancei para uma análise técnica detalhada do código-fonte, produzindo documentos de estudo aprofundado tanto do artigo quanto da implementação, além de criar um quadro comparativo das funções de cada agente e elencar oportunidades de melhoria do sistema, como a adaptação para a B3, a implementação de um módulo de avaliação e a realização de comparações entre as recomendações e os retornos reais. Os materiais produzidos nessas **Semanas 6 e 7** estão consolidados no **Apêndice 3**.

Na **Semana 8**, iniciei a fase de experimentação prática mais intensa com o TradingAgents, buscando entender o fluxo completo de decisão do sistema. Um passo importante foi o desenvolvimento de um web-app em Streamlit para substituir a interface via CLI original. Essa nova interface permitiu visualizar de maneira mais clara as etapas do processo, como chamadas de tools e saídas de cada agente, e se tornou uma ferramenta central para inspeção e depuração do sistema. Além dessa implementação, realizei uma primeira tentativa de replicar o experimento apresentado no artigo, restringindo o escopo à ação da Apple (AAPL) em um período de aproximadamente 65 dias úteis, entre 01/01/2024 e 29/03/2024, utilizando os modelos gpt-4.1 mini (quick-think) e o4-mini (deep-think). Também elaborei uma primeira versão de código para construção de baselines, cálculo de métricas e comparação das estratégias com o TradingAgents. Nesta etapa, observei que os resultados obtidos estavam distantes daqueles reportados na publicação, levantando dúvidas sobre o dataset utilizado pelos autores para realizar os experimentos do artigo e sobre detalhes da metodologia de avaliação. O web-app desenvolvido, os scripts de avaliação e a análise inicial dos resultados da **Semana 8** estão documentados no **Apêndice 4**.

A **Semana 9** foi dedicada a aprofundar a avaliação do sistema e a investigar de forma mais crítica as limitações da abordagem proposta no artigo. Segui o plano de ação definido na Semana anterior, utilizando o LangSmith para monitorar o funcionamento da arquitetura multiagente e confirmar se, ao menos em termos de fluxo e comunicação entre agentes, o comportamento estava alinhado ao descrito pelos autores. Em seguida, explorei o uso de modelos open source rodando localmente via Ollama e vLLM, validando a viabilidade técnica dessa alternativa. Apesar de funcional, a qualidade dos relatórios produzidos pelos modelos de quick think ficou aquém do desejado, o que impactou a cadeia de decisão, enquanto o modelo de deep think (GPT-OSS 20B) apresentou desempenho satisfatório, mas dependente da qualidade das etapas anteriores. Outro ponto importante foi perceber que possuir apenas o arquivo CSV com os sinais de buy, sell ou hold gerados pelo TradingAgents para cada dia não é suficiente para reproduzir a avaliação do artigo, pois faltam detalhes sobre como os baselines foram construídos e como as métricas foram calculadas. Tentativas de esclarecimento via GitHub, Discord e contato via e-mail com os autores não trouxeram respostas conclusivas, o que reforçou as dúvidas sobre a validade

dos resultados publicados. Nesse mesmo período, iniciei a análise de um novo trabalho, “Agent Trading Arena: A Study on Numerical Understanding in LLM-Based Agents”, que propõe um ambiente dinâmico em que agentes competem no mercado financeiro e são testados por sua capacidade estratégica e adaptativa, evidenciando ganhos quando se combinam entradas visuais e módulos de reflexão. Os registros de experimentos, conclusões e notas produzidos na **Semana 9** compõem o **Apêndice 5**.

Na **Semana 10**, dediquei-me a superar a simples replicação e ajustar o código do TradingAgents para buscar um desempenho mais consistente e realista. Uma das principais mudanças foi a eliminação do look-ahead bias nas ferramentas, o que exigiu a construção de conjunto de dados fundamentalistas históricos para garantir que o sistema não acessasse informações futuras durante a simulação. Além disso, refinei os prompts para mitigar os vieses interpretativos dos analistas e fortalecer o debate do Research Team, identificado como a peça central para a qualidade das recomendações. Ao reproduzir as condições experimentais do artigo com os ativos AAPL e GOOGL, os resultados obtidos ficaram abaixo dos números reportados pelos autores, mas superiores aos baselines clássicos, validando a vantagem da arquitetura multiagente sobre um LLM atuando isoladamente. O resultado final demonstrou que agentes baseados puramente em LLMs enfrentam dificuldades em operação intraday, sugerindo que a evolução do projeto deve caminhar para a integração com métodos de séries temporais, o uso de LLMs especializados no mercado financeiro ou a adaptação para estratégias de Swing Trade. A sistematização dos experimentos e análises realizadas na **Semana 10** encontra-se organizada no **Apêndice 6**.

Ao olhar para toda essa jornada, sinto que a Residência em Inteligência Artificial contribuiu de forma única e extremamente significativa para a minha formação, tanto no conhecimento técnico da área de Agentes Inteligentes quanto na minha evolução pessoal. O aprendizado abrangeu desde o contato inicial com a teoria até a execução prática, marcada pela superação dos desafios naturais de um projeto dessa magnitude. Essa vivência foi decisiva para consolidar minha expertise em uma área que, até então, não era de meu domínio. Por fim, gostaria de registrar meu agradecimento aos docentes do Bacharelado de Inteligência Artificial, em especial aos professores Cedric, Fernando Federson e Leonardo,

que conduziram a Residência. Agradeço também aos colegas de turma, que compartilharam comigo cada etapa e aprendizado desta trajetória. Não poderia encerrar sem agradecer à minha família; o suporte emocional e o incentivo contínuo que me ofereceram ao longo desses anos foram pilares fundamentais para o sucesso e a conclusão desta realização.

APÊNDICE 1

Termo de Aceite de Entrega

Objetivo deste documento

Este documento faz parte do Processo da disciplina Residência em IA e tem como objetivo formalizar o aceite da entrega considerando o planejado e o realizado para o período.

Data da Reunião ("Gate") de aprovação: 4 de set. de 2025

Participantes da Entrega [matriculados em Residência em IA]:

ALBERTO LUCAS BORGES DE ALMEIDA TEIXEIRA

Entrega: [descrever a ENTREGA - requisitos e produtos gerados: links para textos, códigos, vídeos etc.]

Durante a primeira Semana de Residência, foram realizadas as seguintes atividades:

- Definição da área em que quero me tornar especialista:
  - Essa decisão foi tomada após analisar os tópicos presentes nos congressos: **ICAI'25**, **ICDATA'25** e **ACC'25**.
  - Os tópicos que mais me chamaram a atenção foram:
    - **Agentes Inteligentes**
    - **Processamento de Linguagem Natural**
    - **Sistemas de Aprendizagem Multimodal**
  - Com base nos tópicos que me interessaram e numa análise inicial de cada um, escolhi **"Agentes Inteligentes"** como foco de aprofundamento na residência.

- Como ponto de partida, iniciei meus estudos com o livro **Inteligência Artificial: Uma Abordagem Moderna (Russell & Norvig)**:
  - Leitura dos capítulos 1 e 2
  - Considerado o livro-texto mais popular da área de IA
  - Os autores introduzem o conceito de Agentes Inteligentes
  - Anotações Artificial Intelligence: A Modern Approach

- Busca de artigos e outros materiais relacionados ao tema:
  - Levantamento de artigos relevantes, que serão estudados no decorrer das próximas semanas
  - Artigos de interesse

Para as atividades dessa semana foram utilizadas as seguintes ferramentas:

- Consensus
- Connected papers
- ChatGPT/Gemini
- NotebookLM

Planejamento: [descrever o que pretende fazer para realizar a próxima ENTREGA]

- Leitura dos 3 primeiros artigos de interesse selecionados, fazendo anotações sobre eles
- Buscar mais artigos que tratem sobre aplicações envolvendo LLM-based Agents

Observação: [caso precise fazer alguma observação, de qualquer "natureza"]

ACEITE DA ENTREGA:

CEDRIC LUIZ DE CARVALHO: Go!

Artificial Intelligence: A Modern Approach (Russell & Norvig) - Anotações

Um **agente** é definido como **qualquer entidade que pode perceber seu ambiente por meio de sensores** e **agir sobre esse ambiente por meio de atuadores**.

Exemplos de Agentes:

- Ser humano: olhos/ouvidos (sensores) - mãos/pernas (atuadores)
- Robô: câmeras (sensores) -motores (atuadores)

O agente percebe o ambiente, processa essa percepção e decide uma ação a ser executada. O comportamento de um agente é formalmente descrito por uma **função de agente**, que mapeia qualquer sequência de percepções a uma ação. Essa função é uma descrição matemática abstrata, enquanto o **programa do agente** é a implementação concreta que roda em uma arquitetura física.

O simples fato de perceber e agir não é suficiente para que um agente seja considerado inteligente. O ponto central é a **racionalidade**. Um **agente racional** é aquele que age para **maximizar o valor esperado de uma medida de desempenho**, com base no conhecimento que possui e nas percepções recebidas até o momento.

- **Racionalidade ≠ onisciência**: não exige perfeição; decide bem com informação limitada.

- **Autonomia e aprendizado**: depender só do projetista não basta; o agente deve **aprender com a experiência** (percepções) para corrigir lacunas do conhecimento inicial.

- **Com experiência suficiente**: o comportamento tende a se tornar **independente** do conhecimento inicial.

Para projetar um agente racional, o primeiro passo é **definir o ambiente da tarefa** por meio do acrônimo **PEAS (Performance, Environment, Actuators, Sensors)**.

• **Performance** (Desempenho): O que constitui sucesso?
Ex: para um táxi, segurança, rapidez, lucro

• **Environment** (Ambiente): Onde o agente opera?
Ex: ruas, tráfego, pedestres, clientes

• **Actuators** (Atuadores): Como o agente age no ambiente?
Ex: volante, acelerador, freio, visor

• **Sensors** (Sensores): Como o agente percebe o ambiente?
Ex: câmeras, GPS, velocímetro

Vale destacar que o **PEAS** se concentra no problema, na definição de metas e restrições, e não na tecnologia. Quando essa etapa é bem executada, é possível **antecipar os *trade-offs*** e como serão medidos, além de facilitar a **avaliação da solução** e a **comparação entre abordagens**.

Tipos de ambientes de tarefa (a "dificuldade" do jogo)

A natureza do ambiente molda diretamente a estratégia do agente. Entender essas dimensões evita soluções incompatíveis com a realidade.

- **Completamente vs. Parcialmente observável**
  - Completamente observável: os sensores capturam o estado completo necessário para decidir (ex.: xadrez em tabuleiro físico bem visto).
  - Parcialmente observável: faltam informações relevantes (ex.: dirigir um táxi, onde não se sabe a intenção dos outros motoristas).

- **Agente único vs. Multiagente**
  - Agente único: o desempenho não depende de outros agentes (ex.: palavras cruzadas).
  - Multiagente: outros agentes influenciam o ambiente (ex.: xadrez, trânsito).
  - Decidir bem implica prever o comportamento alheio e, às vezes, competir ou cooperar.

- **Determinístico vs. Estocástico**

- Determinístico: a mesma ação no mesmo estado gera o mesmo resultado (ex.: aspirador simplificado sem ruído).
- Estocástico: há incerteza nos resultados (ex.: trânsito real, clima). Requer probabilidades, utilidade esperada e/ou políticas robustas.

- **Episódico vs. Sequencial**
  - Episódico: cada decisão é independente das futuras (ex.: classificar uma peça e seguir para a próxima).
  - Sequencial: decisões afetam o futuro (ex.: xadrez, dirigir).

- **Estático vs. Dinâmico**
  - Estático: o mundo não muda enquanto o agente delibera (ex.: palavras cruzadas).
  - Dinâmico: o mundo evolui durante a decisão (ex.: direção).
  - Exige tempo real, replanejamento e reatividade.

- **Discreto vs. Contínuo**
  - Discreto: estados/ações/tempo em passos distintos (ex.: xadrez).
  - Contínuo: variáveis e tempo fluem (ex.: velocidade, posição, direção).

- **Conhecido vs. Desconhecido**
  - Conhecido: as “regras do mundo” (modelos de transição/observação) são conhecidas.
  - Desconhecido: é preciso aprender como o mundo responde (exploração, modelagem, aprendizado por reforço).

A estrutura dos agentes: uma hierarquia de complexidade

- **Agentes reativos simples**
  - Selecionam ações com base **apenas** na percepção atual, via regras:
    - Condição → ação (“se X, então Y”).
  - Vantagens: latência mínima, implementação simples.

- Limites: sem memória.

- **Agentes reativos baseados em modelo**
  - Mantêm **estado interno** para representar aspectos **não observados**; usam um modelo de transição (como o mundo evolui) e, às vezes, um modelo de observação (como percebo o mundo).
  - Resultado: decisões consideram **histórico** e **preenchimento de lacunas**.

- **Agentes baseados em objetivos**
  - Além do modelo, possuem **metas explícitas** (estados desejáveis).
  - Usam **busca/planejamento** para selecionar sequências de ações que **alcançam** essas metas.
  - Ponto forte: **flexibilidade**, trocar a meta muda o comportamento sem reescrever regras.

- **Agentes baseados em utilidade**
  - Quando há **múltiplas metas**, **conflitos** ou **incerteza**, utilizam uma **função de utilidade** para quantificar preferências (quão "bom" é um estado/resultado).
  - Decidem maximizar a **utilidade esperada**.
  - Formaliza **trade-offs** (ex.: rapidez vs. segurança) e **avalia risco**.

- **Agentes com aprendizagem (meta-agentes)**
  - Qualquer um dos tipos acima pode incorporar **aprendizado** para melhorar no tempo.
  - Quatro elementos:
    • **Elemento de desempenho**: executa a política atual (atua).
    • **Elemento de aprendizado**: ajusta parâmetros/modelos para **melhorar**.
    • **Crítico**: **avalia** o comportamento com base em feedback/medida de desempenho.
    • **Gerador de problemas**: propõe **novas experiências** (exploração) que aceleram o aprendizado.
  - Efeito: operam em **ambientes desconhecidos**, **adaptam-se** a mudanças e **evoluem** competência com experiência.

- Sobre a estrutura dos agentes é possível sumarizar que:
  - Um agente reativo simples é ótimo para **tarefas estáveis** e bem observáveis; e não desempenha tão bem em ambientes **parciais/dinâmicos**.
  - Estado interno + modelo **estabilizam** decisões sob percepção imperfeita.
  - Objetivos dão **direção**.
  - Utilidade dá **critério** sob conflito e incerteza.
  - Sem o aprendizado, o agente fica **dependente** do projetista; com aprendizado, ganha **autonomia** e melhora contínua.

Artigos de Interesse para a Residência

Este documento será atualizado continuamente com novos materiais, servindo como guia para orientar meus estudos ao longo do Processo da Residência

- **Materiais introdutórios para compreensão geral do tema:**
  - **Inteligência Artificial: Uma Abordagem Moderna (Russell & Norvig)**
    - Link: Inteligência Artifial Russel & Norvig - Capítulos 1 e 2.pdf
  - **Intelligent Agents: Theory and Practice**
    - Link: woodridge_intelligent_agents.pdf

- **Surveys sobre LLM-based agents:**
  - **A Survey on Large Language Model based Autonomous Agents**
    - Link: **[2308.11432] A Survey on Large Language Model based Autonomous Agents**
  - **Large Language Model Agent: A Survey on Methodology, Applications and Challenges**
    - Link: **[2503.21460] Large Language Model Agent: A Survey on Methodology, Applications and Challenges**
  - **A Survey of LLM-based Agents: Theories, Technologies, Applications and Suggestions (pdf baixado no mac)**
    - Link: **A Survey of LLM-based Agents: Theories, Technologies, Applications and Suggestions | IEEE Conference Publication | IEEE Xplore**

- **Alguns artigos que chamaram minha atenção durante a busca por *surveys* de LLM-based agents. Vou mantê-los salvos aqui para uma análise posterior, pois podem ser úteis.**

- **Large Multimodal Agents: A Survey**
  - Link: **2402.15116**
- **A Survey onThe Memory Mechanism of Large Language Models based Agents:**
  - Link: **2404.13501**
- **A Survey on the Optimization on Large Language Model-based Agents**
  - Link: **A Survey on the Optimization of Large Language Model-based Agents**
- **Multi-Agent Collaboration Mechanisms: A Survey of LLMs**
  - Link: **Multi-Agent Collaboration Mechanisms: A Survey of LLMs**
- **A Survey of Large Language Model Agents for Question Answering:**
  - Link: **2503.19213**
- **Large Language Model Agent in Financial Trading: A Survey**
  - Link: Large Language Model Agent in Financial Trading: A Survey

Termo de Aceite de Entrega

Objetivo deste documento

Este documento faz parte do Processo da disciplina Residência em IA e tem como objetivo formalizar o aceite da entrega considerando o planejado e o realizado para o período.

Data da Reunião ("Gate") de aprovação: 10 de set. de 2025

Participantes da Entrega [matriculados em Residência em IA]:

ALBERTO LUCAS BORGES DE ALMEIDA TEIXEIRA

Entrega: [descrever a ENTREGA - requisitos e produtos gerados: links para textos, códigos, vídeos etc.]

Durante a segunda Semana do Processo da Residência, foram realizadas as seguintes atividades:

- Leitura do artigo **Intelligent Agents: Theory and Practice (Michael Wooldridge & Nicholas R. Jennings)**
  - Um dos trabalhos mais influentes e citados sobre agentes inteligentes.
  - Neste artigo são explorados os conceitos fundamentais, as diretrizes de projeto, opções de arquitetura e os principais desafios.
  - Anotações - Intelligent Agents: Theory and Practice

- Estabelecimento de um **método** para seleção de materiais:

  - A proposta é uma metodologia de escolha de artigos **inspirada** em como é feita uma Revisão de Literatura, **mas adaptada ao objetivo da Residência**: me tornar um especialista em uma área (Agentes Inteligentes).
  - A diferença principal é que, em vez de analisar a literatura científica a fim de identificar lacunas e justificar novas pesquisas, **o meu objetivo com esse método é escolher bons materiais** para embasar os meus estudos nessa fase de especialização.
  - Inicialmente, concentrei-me na trajetória e nos fundamentos de agentes de forma ampla. Para afunilar o escopo, passarei a priorizar **Agentes baseados em LLM**.
  - Como ponto de partida, optei por mapear **surveys** relevantes para construir uma **visão ampla da área**. A partir daí, consigo direcionar melhor a escolha e a prioridade dos eixos temáticos.
  - Os sites de buscas utilizados foram:
    - ACM Digital Library;
    - arXiv;
    - IEEE;
  - Critérios utilizados:
    - Uso dos mecanismos de busca avançada:
      - Termos de interesse;

- Data;
  - Seleção inicial mais ampla;
  - Refinamento da seleção após a leitura do abstract e conclusão de cada artigo;
- **Resultado final:** 2 artigos selecionados pelo método e 1 artigo indicado por um colega.
- Com os materiais de interesse selecionados, revisei o documento da Semana 1, aprimorando-o com uma seleção mais precisa e filtrada de artigos, para que sirva, daqui em diante, como base para orientar melhor meus estudos.
- Artigos de interesse

Planejamento: [descrever o que pretende fazer para realizar a próxima ENTREGA]

- **Leitura dos Surveys selecionados:** ler os materiais priorizados, marcar trechos-chave e produzir um resumo objetivo.
- **Extração de tópicos:** listar os principais temas citados nos surveys que servem como eixos de aprofundamento.
- **Planejamento da próxima etapa:** estruturar um cronograma dos temas escolhidos e sua ordem de aprofundamento

Observação: [caso precise fazer alguma observação, de qualquer "natureza"]

O artigo indicado foi **"Large Language Model Agent: A Survey on Methodology, Applications and Challenges"**, sugerido pelo Amir.

ACEITE DA ENTREGA:

CEDRIC LUIZ DE CARVALHO: Go!

Intelligent Agents: Theory and Practice (Wooldridge & Jennings) - Anotações

Os autores dividem o campo em três áreas centrais:

- **Teorias de agentes**, que se preocupam com a formalização e as propriedades dos agentes.
- **Arquiteturas de agentes**, que são os modelos de engenharia para sua construção.
- **Linguagens de agentes**, que são as ferramentas de programação para implementá-los.

O Problema da definição de "Agente":

Um ponto de partida do artigo é a dificuldade em se estabelecer uma definição única e universalmente aceita para o termo "agente". Os autores propõem uma distinção entre duas noções principais:

- **Noção Fraca:** descreve um agente como um sistema de hardware ou software que possui quatro propriedades essenciais:
    - **Autonomia:** agentes operam sem a intervenção direta de humanos e têm controle sobre suas próprias ações e estado interno.
    - **Habilidade Social:** agentes interagem com outros agentes (e possivelmente humanos) por meio de uma linguagem de comunicação.
    - **Reatividade:** agentes percebem seu ambiente e respondem de forma oportuna às mudanças que nele ocorrem.
    - **Proatividade:** agentes não apenas reagem ao ambiente, mas são capazes de exibir comportamento direcionado a objetivos, tomando a iniciativa.

- **Noção Forte:** utilizada principalmente em IA, esta noção adiciona à definição anterior conceitos geralmente aplicados a humanos. Um agente é caracterizado a partir de noções mentalistas como **conhecimento, crença e intenção**. Essa abordagem, chamada de **"postura intencional"** por Daniel Dennett, é vista como

uma poderosa ferramenta de abstração para descrever e prever o comportamento de sistemas complexos.

Os Três Pilares da Pesquisa em Agentes

O campo é estruturado em torno de três áreas de investigação que vão da especificação à implementação.

1. **Teorias de Agentes:** funcionam como a especificação formal. Buscam definir o que um agente é e como ele deve se comportar racionalmente. Utilizam lógicas modais para modelar atitudes como conhecimento e crença. A abordagem mais comum é a **semântica de mundos possíveis**, que define a crença de um agente como aquilo que é verdade em todos os mundos que ele considera possíveis.
    a. **Grande Desafio:** essa abordagem sofre do **problema da onisciência lógica**, pois assume que os agentes são raciocinadores perfeitos, capazes de conhecer todas as tautologias e derivar todas as consequências lógicas de suas crenças, o que é computacionalmente irrealista.
    b. **Teorias Importantes:** a teoria da **intenção como um objetivo persistente**, de Cohen e Levesque, e as arquiteturas lógicas **BDI (Beliefs, Desires, Intentions)**, de Rao e Georgeff, são exemplos de tentativas de formalizar a relação entre as atitudes mentais para explicar a ação racional.

2. **Arquiteturas de Agentes:** são os blueprints de engenharia que descrevem como construir um agente na prática, mapeando percepções (sensores) em ações. Existem três paradigmas principais:
    a. **Deliberativas (IA Simbólica):** possuem um modelo simbólico explícito do mundo e tomam decisões através de raciocínio lógico e planejamento (ex: sistemas baseados em STRIPS). Seu ponto fraco é a complexidade computacional, tornando-os lentos para ambientes dinâmicos.
    b. **Reativas:** rejeitam o modelo simbólico central. A inteligência "emerge" da interação direta com o ambiente. O exemplo clássico é a **Arquitetura de Subsunção** de Rodney Brooks, onde comportamentos são organizados em camadas hierárquicas, com as mais baixas (e mais primitivas) tendo precedência.
    c. **Híbridas:** buscam combinar o melhor dos dois mundos. Geralmente são estruturadas em camadas, com um componente reativo para respostas rápidas e um componente deliberativo para planejamento de longo prazo. O

PRS (Procedural Reasoning System), que implementa o modelo BDI, e a **TURING MACHINES** são exemplos notáveis.

3. **Linguagens de Agentes:** são as ferramentas de programação que implementam os conceitos teóricos. A ideia é programar um sistema diretamente em termos de noções mentalistas. Exemplos incluem:
    a. **AGENT0** (de Yoav Shoham), que propõe o paradigma da Programação Orientada a Agentes (AOP).
    b. **Concurrent METATEM**, onde o comportamento do agente é especificado em lógica temporal e executado diretamente.

Aplicações da Tecnologia de Agentes

O artigo finaliza mostrando que a tecnologia de agentes não é puramente teórica, possuindo aplicações em diversas áreas:

- **Controle e Logística:** gerenciamento de tráfego aéreo, controle de processos industriais e redes de telecomunicações.
- **Agentes de Interface:** assistentes pessoais que aprendem as preferências do usuário para filtrar informações (ex: e-mails, notícias).
- **Agentes de Informação:** sistemas que buscam e integram dados de múltiplas fontes na internet ou em bancos de dados heterogêneos.
- **Entretenimento:** agentes "verossímeis" (believable agents) em jogos e realidade virtual, que precisam exibir emoções e comportamentos críveis para criar uma "ilusão de vida".

APÊNDICE 2

Termo de Aceite de Entrega

Objetivo deste documento

Este documento faz parte do Processo da disciplina Residência em IA e tem como objetivo formalizar o aceite da entrega considerando o planejado e o realizado para o período.

Data da Reunião ("Gate") de aprovação: 17 de set. de 2025

Participantes da Entrega [matriculados em Residência em IA]:

ALBERTO LUCAS BORGES DE ALMEIDA TEIXEIRA

Entrega: [descrever a ENTREGA - requisitos e produtos gerados: links para textos, códigos, vídeos etc.]

Durante a terceira Semana da Residência, foram realizadas as seguintes atividades:

- Recapitulando a minha estratégia:
  - Leitura dos capítulos iniciais do livro: **Inteligência Artificial: Uma Abordagem Moderna (Russell & Norvig)**
  - Leitura do artigo: **Intelligent Agents: Theory and Practice (Michael Wooldridge & Nicholas R. Jennings)**
  - Migrar para LLM-based Agents (atual)

- Estudo do Survey: **A Survey on Large Language Model based Autonomous Agents**
  - Dividido em 3 aspectos chave: **construção**, **aplicação** e **avaliação**
  - **Framework unificado:** perfil, memória, planejamento e ação
  - Como dar capacidade aos agentes
  - Principais gargalos e o futuro da área
  - Após a leitura, elaborei um resumo com os principais pontos para revisões futuras:
  - Anotações - A Survey on Large Language Model based Autonomous Agents

- Extração dos tópicos:
  - Após a leitura do survey, conforme planejado, ampliei minha compreensão da área e pude definir subáreas em agentes baseados em LLM que orientarão meus estudos daqui em diante
  - Subáreas de LLM-based agents

Planejamento: [descrever o que pretende fazer para realizar a próxima ENTREGA]

- **Busca artigos** usando o método elaborado na Semana passada (inspirado em revisão de literatura).
  - **Objetivo:** registrar, no documento criado esta Semana, os artigos selecionados por subárea.

- **Início da leitura** em uma das subáreas, dando partida no aprofundamento desse tema.
    - Gerar materiais de apoio a partir da leituras

Observação: [caso precise fazer alguma observação, de qualquer "natureza"]

ACEITE DA ENTREGA:

CEDRIC LUIZ DE CARVALHO: Go!

A Survey on Large Language Model based Autonomous Agents - Anotações

A Arquitetura do Agente

A maioria dos agentes de IA segue uma estrutura unificada com quatro componentes principais que trabalham juntos:

- **Módulo de Perfil (A Identidade do Agente):** este módulo define quem o agente é (um programador, um cientista, um personagem extrovertido, etc.), o que molda todo o seu comportamento. A criação desse perfil pode ser feita de três formas:
    - **Manual:** desenvolvedores definem o perfil diretamente com instruções como "Você é um especialista em desenvolvimento de software".
    - **Gerada por IA:** um LLM, cria novos perfis com base em alguns exemplos iniciais, automatizando o processo.
    - **Baseado em Dados Reais:** perfis são extraídos de dados do mundo real (como pesquisas demográficas) para garantir que o agente se comporte de maneira mais realista e representativa de uma população específica.

- **Módulo de Memória:** a memória é crucial para que o agente aprenda com o passado e mantenha a consistência. Ela funciona de forma parecida com a memória humana, com componentes de curto e longo prazo.
    - **Estruturas de Memória:**
        - **Memória Unificada (Curto Prazo):** o agente lembra apenas das informações mais recentes, contidas no contexto da conversa atual. É útil para ações imediatas.
        - **Memória Híbrida (Curto e Longo Prazo):** combina a memória de curto prazo com um banco de dados externo (a memória de longo prazo). Isso permite que o agente "lembre" de experiências passadas, mesmo que tenham ocorrido há muito tempo, recuperando informações relevantes quando necessário.
    - **Operações de Memória:**
        - **Leitura:** o agente busca em sua memória as informações mais importantes para a tarefa atual, considerando o quão recente, relevante e significativa é cada lembrança.

    - **Escrita:** armazena novas informações, tomando cuidado para não duplicar dados e descartando memórias antigas ou menos importantes para evitar sobrecarga.
    - **Reflexão:** o agente analisa suas memórias de baixo nível (observações simples) para gerar conclusões de alto nível. Por exemplo, ao observar que uma pessoa está sempre lendo e falando sobre sua pesquisa, o agente pode refletir e concluir que "essa pessoa é dedicada ao seu trabalho".

- **Módulo de Planejamento (A Estratégia do Agente):** este módulo permite que o agente divida tarefas grandes e complexas em passos menores e mais fáceis de gerenciar.
    - **Planejamento sem Feedback:** o agente cria um plano completo de uma só vez e o executa sem fazer ajustes. Um exemplo é a técnica de "Cadeia de Pensamento" (Chain of Thought), onde a IA é instruída a "pensar passo a passo".

    - **Planejamento com Feedback:** o agente ajusta seu plano conforme age e recebe novas informações. O feedback pode vir de diferentes fontes:
        - **Do Ambiente:** o agente executa uma ação, observa o resultado e usa essa observação para planejar o próximo passo.
        - **De um humano:** uma pessoa pode guiar ou corrigir o plano do agente.
        - **Do Próprio Modelo (Autorreflexão):** o agente avalia seu próprio desempenho, identifica erros e se corrige, aprendendo com a própria experiência.

- **Módulo de Ação:** é aqui que o agente executa as tarefas, transformando decisões em ações concretas.
    - **Fontes de Ação:** as ações podem vir de ferramentas externas (como usar uma API para buscar dados na internet ou consultar um banco de dados) ou do conhecimento interno do próprio LLM (como conversar, raciocinar e usar o bom senso).

    - **Impacto da Ação:** uma ação pode alterar o ambiente (como pegar um item em um jogo), modificar o estado interno do agente (atualizar sua memória) ou simplesmente ser o gatilho para a próxima ação em uma sequência.

Adquirindo Habilidades

Além da estrutura, o agente precisa de habilidades. Elas podem ser adquiridas de duas maneiras:

- **Com Fine-tuning**: este método envolve treinar o LLM com dados específicos para especializá-lo em uma tarefa. É como dar a um estudante geral um curso de especialização. Os dados de treinamento podem vir de anotações feitas por humanos, de outros LLMs ou de exemplos do mundo real.

- **Sem Fine-tuning:** as habilidades são aprimoradas sem alterar o modelo original, usando técnicas inteligentes de interação.
    - **Engenharia de Prompt:** criar instruções (prompts) muito bem elaboradas para extrair o máximo da capacidade do LLM. A instrução "pense passo a passo" é um exemplo simples que melhora drasticamente o raciocínio.
    - **Engenharia de Mecanismos:** desenvolver sistemas externos que ajudam o agente. Isso inclui mecanismos como:
        - **Tentativa e Erro:** o agente tenta algo, recebe uma crítica e se corrige.
        - **Debate entre Agentes:** vários agentes discutem um problema para encontrar a melhor solução.
        - **Acúmulo de Experiência:** o agente guarda soluções bem-sucedidas para reutilizá-las no futuro.

Aplicações em Múltiplos Domínios

Áreas citadas nos Survey:

- **Ciências Sociais:** agentes simulam comportamentos humanos para estudar psicologia, ciência política e fenômenos sociais em ambientes virtuais. Eles também atuam como assistentes de pesquisa, resumindo artigos, e como assistentes legais, analisando casos.
- **Ciências Naturais**: eles aceleram a pesquisa científica ao extrair dados de artigos, gerenciar bancos de dados químicos e até mesmo planejar e executar experimentos em laboratório. Também funcionam como tutores para estudantes de matemática e programação.
- **Engenharia:**
    - **Desenvolvimento de Software:** equipes de agentes colaboram para automatizar todo o ciclo de desenvolvimento: escrevem código, testam, corrigem bugs e criam a documentação.

- **Automação Industrial:** controlam processos de produção de forma inteligente e flexível.
- **Robótica:** permitem que robôs entendam comandos complexos em linguagem natural e os transformem em uma sequência de ações físicas.

Avaliação de um Agente

- **Avaliação Subjetiva (julgamento humano):** utilizada quando não há uma resposta certa ou errada, focando na qualidade da interação.
  - **Anotação Humana:** especialistas dão notas ao desempenho do agente, avaliando se ele foi coerente, útil ou se manteve no personagem.
  - **Teste de Turing:** avaliadores tentam adivinhar se estão interagindo com um humano ou com o agente. Se não conseguirem diferenciar, o agente é considerado bem-sucedido.

- **Avaliação Objetiva (Métricas e Testes):** usa números e dados concretos para medir o desempenho de forma padronizada.
  - **Métricas de Desempenho:** medem a taxa de sucesso em uma tarefa, a eficiência (custo e tempo) e o quão parecido o comportamento do agente é com o de um humano.
  - **Benchmarks:** são ambientes e tarefas padronizadas (como um jogo ou um site de compras simulado) que permitem comparar o desempenho de diferentes agentes de forma justa.

Principais Desafios e o Futuro da área

- **Simulação de Papéis Específicos:** os agentes ainda têm dificuldade em interpretar papéis muito específicos ou que exigem um conhecimento profundo da psicologia humana.
- **Alinhamento de Valores:** para simulações realistas, os agentes precisam representar uma gama diversa de personalidades, incluindo traços negativos, o que é um desafio de alinhar de forma segura.
- **Fragilidade dos Prompts:** o comportamento dos agentes depende de instruções complexas. Pequenas mudanças nessas instruções podem gerar resultados totalmente diferentes e instáveis.
- **Alucinação:** assim como outros LLMs, os agentes podem inventar informações falsas e apresentá-las com grande confiança, o que é perigoso em aplicações críticas.

- **Limitação de Conhecimento:** é difícil fazer com que um agente "esqueça" o vasto conhecimento da internet para que ele aja como um humano com conhecimento limitado em uma simulação.
- **Eficiência:** como os agentes precisam consultar o LLM várias vezes para planejar, lembrar e agir, seu funcionamento pode ser lento e custoso.

Subáreas de agentes baseados em LLM que pretendo estudar com mais profundidade

Otimização

- A Survey on the Optimization of Large Language Model-based Agents (Link: A Survey on the Optimization of Large Language Model-based Agents)

Memória (Arquitetura)

- A Survey on the Memory Mechanism of Large Language Model based Agents (Link: 240.413.501)

Aplicação

- TradingAgents: Multi-Agents LLM Financial Trading Framework (Link: Large Language Model Agent in Financial Trading: A Surveyl)

Segurança

- AI Agents Under Threat: A Survey of Key Security Challenges and Future Pathways (Link: AI Agents Under Threat: A Survey of Key Security Challenges and Future Pathwayst)
- The Emerged Security and Privacy of LLM Agent: A Survey with Case Studies (Link: The Emerged Security and Privacy of LLM Agent: A Survey with Case Studies)

Avaliação

- Survey on Evaluation of LLM-based Agents (Link: 250.316.416)

Multiagente

- BeyondSelf-Talk: A Communication-CentricSurveyofLLM-Based Multi-Agent Systems (Link: Beyond Self-Talk: A Communication-Centric Survey of LLM-Based Multi-Agent Systems)
- A Survey on LLM-based Multi-Agent System: Recent Advances and New Frontiers in Application (Link: 241.217.481

- LLMs Working in Harmony: A Survey on the Technological Aspects of Building Effective LLM-Based Multi Agent Systems (Link: 250.401.963)

Planejamento (Arquitetura)

- Understanding the planning of LLM agents: A survey (Link: 240.202.716)

Termo de Aceite de Entrega

Objetivo deste documento

Este documento faz parte do Processo da disciplina Residência em IA e tem como objetivo formalizar o aceite da entrega considerando o planejado e o realizado para o período.

Data da Reunião ("Gate") de aprovação: 25 de set. de 2025

Participantes da Entrega [matriculados em Residência em IA]:

ALBERTO LUCAS BORGES DE ALMEIDA TEIXEIRA

Entrega: [descrever a ENTREGA - requisitos e produtos gerados: links para textos, códigos, vídeos etc.]

Durante a quarta Semana do Processo da Residência (especialização em Agentes Inteligentes), foram realizadas as seguintes atividades:

- Busca de artigos utilizando o **método** elaborado na segunda Semana:
  - **Objetivo:** selecionar artigos para aprofundar minha especialização em agentes baseados em LLMs, focando nas **"subáreas"** selecionadas na semana passada.
    - **Subáreas**: **Otimização** (aquisição de capacidade: fine-tuning, engenharia de prompt); **Segurança**; **Multiagentes**; **Avaliação**; **Arquitetura** (memória, planejamento); **Aplicação** (não é o foco por enquanto)
  - **Recapitulando o método:**
    - Pesquisa avançada;
    - Seleção inicial mais ampla;
    - Refinamento da seleção após a leitura do abstract e da conclusão;
  - **Resultado final:** pelo menos um artigo (survey) de cada "subárea" que eu julgo importante o aprofundamento.
  - **Organização:** migrei os artigos do docs para uma planilha, melhorando a organização e o acompanhamento dos estudos.
  - Artigos - LLM based agents
- Estudo do Survey: **A Survey on the Optimization of Large Language Model-based Agents**
  - Após a leitura, elaborei um resumo dos pontos-chave para revisões futuras e um eventual aprofundamento.
  - Anotações - A Survey on the Optimization of Large Language Model-based Agents

Planejamento: [descrever o que pretende fazer para realizar a próxima ENTREGA]

- Manter o planejamento em curso e avançar nas leituras por subárea

- Para a próxima Semana, pretendo estudar os seguintes artigos:
    - **A Survey on the Memory Mechanism of Large Language Model based Agents**
    - **Understanding the planning of LLM agents: A survey**
    - **Survey on Evaluation of LLM-based Agents**

Observação: [caso precise fazer alguma observação, de qualquer "natureza"]

ACEITE DA ENTREGA:

CEDRIC LUIZ DE CARVALHO: Go!

A Survey on the Optimization of Large Language Model-based Agents - Anotações

1) Contexto e objetivo

O artigo parte do avanço recente dos LLMs e do uso crescente de agentes autônomos em tarefas interativas e de decisão. Defende que **otimizar agentes é diferente de apenas "melhorar um LLM"**: é preciso aprimorar planejamento de múltiplos passos, memória útil e adaptação a ambientes dinâmicos. A otimização desses agentes podem ser caracterizadas em dois tipos:

- **Parametrizada**: altera pesos do modelo
- **Não parametrizada**: melhora o *uso* do modelo sem mexer nos pesos (memória/experiência, feedback/reflexão, ferramentas, RAG, colaboração multiagente).

2) Fundamentos conceituais

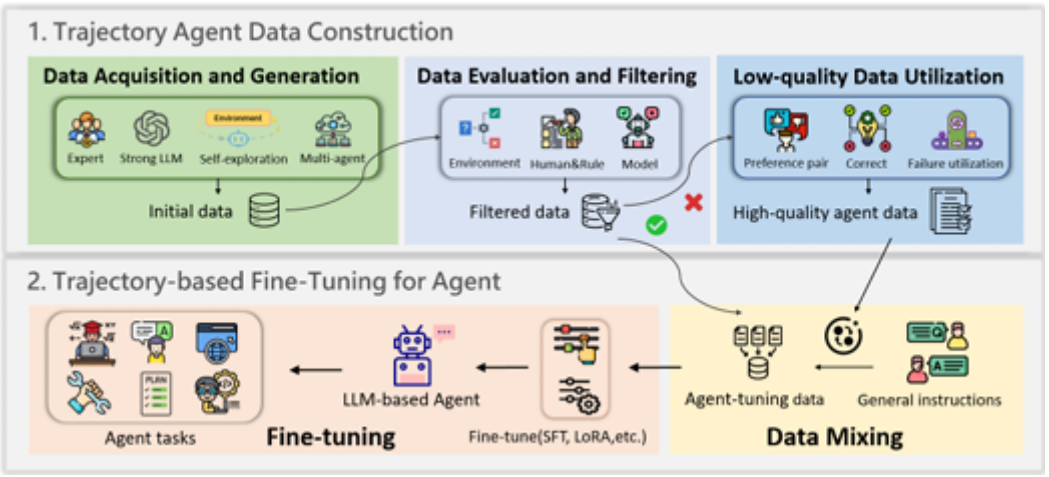
Aprendizagem por Reforço (RL):
Evolui de métodos por valor (p.ex., Q-learning) para métodos por política (PG/PPO). RLHF e DPO entram como formas de alinhar políticas a preferências humanas ou de especialistas. No contexto de agentes, RL oferece o mecanismo para aprender com interação real no ambiente.

Fine-tuning de LLMs:
SFT/Instruction Tuning e PEFT (LoRA/QLoRA) permitem adaptar modelos a tarefas ou estilos, com custo computacional moderado. Em agentes, o uso típico é clonagem de comportamento a partir de trajetórias de qualidade.

RAG (Retrieval Augmented Generation):
Da recuperação "naive" ao RAG modular, a ideia é mitigar conhecimento estático do pré-treino, conectando o agente a bases externas atualizadas e filtradas.

3) Otimização parametrizada

A pergunta aqui é: **como transformar um LLM em um bom agente para tarefas complexas?** O eixo central é **treinar com dados de trajetória** (estado-ação-observação) e refinar a política.



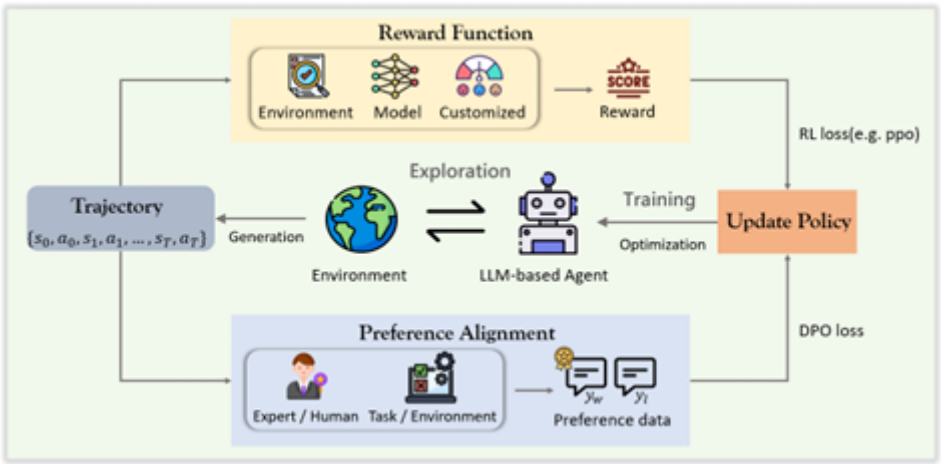
3.1 Fine-tuning convencional

- **Dados de trajetória**: podem vir de (i) especialistas, (ii) LLMs mais fortes, (iii) autoexploração, (iv) colaboração multiagente.
    - Especialistas: alta precisão, porém caro e escasso.
    - LLMs fortes: qualidade estável, mas custo e possíveis vieses.
    - Autoexploração: barato e escalável; exige **filtro forte** para evitar ruído.
    - Multiagente: divide e especializa, mas aumenta complexidade de coordenação.
- **Avaliação/filtragem**: por ambiente (sinal binário de sucesso), por regras/humanos (mais confiável), ou por modelos (automatiza, porém depende do avaliador).
- **Uso de falhas**: amostras ruins não são descartadas; viram pares comparativos (ganhador/perdedor) ou são corrigidas para robustez.
- **Técnicas**:

- **Mistura de dados** (trajetórias + instruções gerais) para não perder capacidades amplas.
- **SFT completo** ou **PEFT** (LoRA/QLoRA) para reduzir custo.
- **Perdas/customizações** específicas da tarefa.

Limite estrutural: depende de dados estáticos; adaptação a cenários não vistos é mais difícil.

3.2 RL para agentes



- **Recompensa explícita**: PPO/Actor–Critic com sinais do ambiente, de um modelo avaliador ou de funções compostas.
- **Alinhamento por preferências (DPO)**: dispensa modelagem de recompensa; usa **pares preferidos** (vencedor vs. perdedor) e otimiza a política offline.
  Útil e simples, mas, isoladamente, tende a favorecer passos únicos; para **tarefas multi-etapas**, combinações são comuns.

3.3 Híbridos (SFT → RL)

Começa com **SFT** para estabilizar o comportamento (inicialização) e segue com **RL**

(PPO/DPO) para refinar por interação. Versões **iterativas** alternam SFT↔RL para ganhos progressivos.

4) Otimização não parametrizada

Foco em como orquestrar o modelo: estrutura de prompt, contexto, memória e ferramentas.

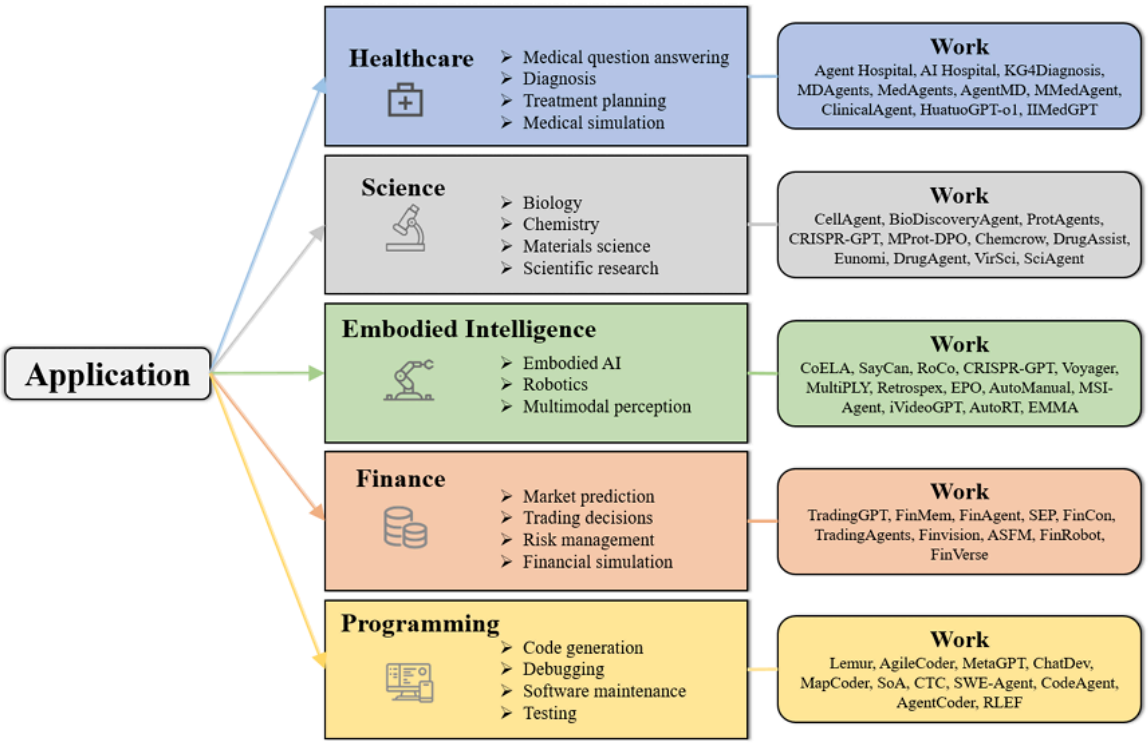
- **Baseada em experiência/memória**: consolida interações passadas para melhor planejamento e reutilização de soluções.
- **Baseada em feedback**:
    - **Autorreflexão** (*self-reflection*): o próprio agente gera críticas/patches textuais.
    - **Feedback externo**: modelos/avaliadores terceiros criticam e o agente revisa.
    - **Otimização de meta-prompts**: ajusta instruções globais a partir de resultados.
- **Baseada em ferramentas**: ensina o agente a escolher e acionar calculadoras, intérpretes e APIs no momento certo (impacto grande em precisão e eficiência).
- **Baseada em RAG**: acopla recuperação seletiva (com ranqueamento/filtragem) e geração; variantes integram reflexão para melhorar a pertinência da busca.
- **Colaboração multiagente**: delega subtarefas a perfis especializados (análise, planejamento, execução), com protocolos de comunicação.

5) Avaliação e aplicações

Benchmarks de avaliação:
Testam **raciocínio matemático** (GSM8K), código (SWE-bench), **web/uso de ferramentas** (WebShop, AgentBench/AgentEval). O artigo valoriza **conjuntos multi-tarefa** que cobrem mais de um eixo (planejamento, ferramentas, web, etc.).

Datasets de *tuning* para agentes:
Trajetórias curadas (pensamento-ação-observação) como **AgentInstruct** e **AgentBank** favorecem SFT/PEFT e DPO, sobretudo em tarefas de programação, matemática, web e IA corporificada.

6) Aplicações



Essas aplicações demonstram impacto concreto, especialmente quando há **RAG** + **ferramentas + memória** e, em muitos cenários, **multiagência**.

7) Desafios e direções futuras

- **Viés e curadoria de dados**: filtros automáticos podem simplificar demais os casos; é preciso fechar o **gap treino-implantação**.

- **Eficiência vs. adaptabilidade**: PPO é caro mas versátil; DPO é leve mas tende ao passo único. Precisamos de **algoritmos eficientes para longas cadeias**.

- **Generalização entre domínios**: agentes afinados num domínio falham ao migrar; urge explorar **transferência e meta-aprendizagem** em nível de agente.

- **Métricas padronizadas**: falta **convergência de métricas** que capturem planejamento, uso de ferramentas e sucesso real em tarefas.

- **Otimização parametrizada em cenários multi-agente**: pouco explorada; há espaço para **métodos de ajuste de políticas coletivas** e coordenação aprendida.

8) Conclusão

- Sem trajetórias de qualidade e sinais de avaliação confiáveis, não existe agente competitivo.
- Sem orquestração cuidada (memória, ferramentas, recuperação e colaboração), o agente não se adapta ao mundo real.

Termo de Aceite de Entrega

Objetivo deste documento

Este documento faz parte do Processo da disciplina Residência em IA e tem como objetivo formalizar o aceite da entrega considerando o planejado e o realizado para o período.

Data da Reunião ("Gate") de aprovação: 2 de out. de 2025

Participantes da Entrega [matriculados em Residência em IA]:

ALBERTO LUCAS BORGES DE ALMEIDA TEIXEIRA

Entrega: [descrever a ENTREGA - requisitos e produtos gerados: links para textos, códigos, vídeos etc.]

Recapitulando:

- Área escolhida: **Agentes Inteligentes**
- Nas primeiras semanas, realizei um estudo mais amplo, consolidando Fundamentos e Teoria de Agentes.
- Depois, avancei para surveys mais generalistas sobre LLM-based agents.
- A partir desse entendimento, selecionei artigos mais focados nos temas identificados nos surveys, esta é minha segunda Semana nessa etapa.

Durante a quinta Semana do Processo de Residência, foram realizadas as seguintes atividades:

- Desde o último Gate fiquei com uma fala do Federson na cabeça: **"Fique tranquilo, ainda há tempo. Mas, ao escolher o rumo (por exemplo, quais artigos ler para definir o tema), considere o que já te tocou mais. Provavelmente houve algo que mexeu mais com o seu coração, leve isso em conta"**
  - A partir disso, cheguei a conclusão de que a escolha dos artigos que eu tinha selecionado para ler essa semana, não levavam em conta isso.
  - Por isso, decidi alterar o planejamento desta semana: reordenei as prioridades de leitura, mas seguirei usando os artigos que já havia selecionado.

- Estudo do Survey: **Understanding the planning of LLM agents: A survey**
  - Anotações - Understanding the planning of LLM agents: A survey

- Estudo do Survey: **LLMs Working in Harmony: A Survey on the Technological Aspects of Building Effective LLM-Based Multi Agent Systems**
  - Anotações - LLMs Working in Harmony: A Survey on the Technological Aspects of Bu…

Planejamento: [descrever o que pretende fazer para realizar a próxima ENTREGA]

- Leitura do Survey: **BeyondSelf-Talk: A Communication-Centric Survey of LLM-Based**

Multi-Agent Systems

- Leitura do Survey: **A Survey on LLM-based Multi-Agent System: Recent Advances and New Frontiers in Application**
- Começar o estudo dos **Frameworks**.

Observação: [caso precise fazer alguma observação, de qualquer "natureza"]

ACEITE DA ENTREGA:

CEDRIC LUIZ DE CARVALHO: Go!

Anotações - Understanding the planning of LLM agents: A survey

O artigo apresenta uma análise concisa das principais metodologias de planejamento para agentes de Inteligência Artificial baseados em Modelos de Linguagem de Grande Escala (LLMs). O uso de LLMs como núcleo cognitivo representa uma mudança em relação às abordagens tradicionais, como o planejamento simbólico e o aprendizado por reforço (RL). Enquanto os métodos simbólicos são rígidos e exigem grande esforço de especialização humana, o RL é notório por sua ineficiência de amostras. Os LLMs surgem como uma alternativa promissora, aproveitando sua capacidade de raciocínio e compreensão de linguagem natural para gerar planos de ação complexos.

Principais Metodologias de Planejamento Baseadas em LLM

As pesquisas atuais sobre o tema podem ser categorizadas em cinco direções principais, cada uma com sua própria lógica e formulação para resolver o problema de gerar uma sequência de ações para atingir um objetivo de um ambiente.

- **1. Decomposição de Tarefas:** Baseada no princípio de "dividir para conquistar", esta abordagem decompõe uma tarefa complexa em subtarefas mais simples e gerenciáveis. A decomposição pode ocorrer de duas formas:
  - **Inicial (Decomposition-First):** O plano é totalmente decomposto no início. Embora garanta o alinhamento com o objetivo final, essa abordagem é frágil, pois um erro em uma etapa inicial pode invalidar todo o plano.
  - **Entrelaçada (Interleaved):** A decomposição e a execução das subtarefas são alternadas, permitindo ajustes dinâmicos com base no feedback do ambiente. Essa flexibilidade aumenta a tolerância a falhas, mas corre o risco de desviar-se do objetivo original em tarefas muito longas (alucinação).

- **2. Seleção de Múltiplos Planos:** Para mitigar a natureza estocástica dos LLMs, esta metodologia gera vários planos candidatos para uma mesma tarefa e, em seguida, utiliza um processo de busca ou votação para selecionar o mais promissor. Abordagens como Tree-of-Thought (ToT) exploram sistematicamente o espaço de soluções, aumentando a chance de encontrar um plano robusto. A principal

desvantagem é o alto custo computacional, já que a geração e avaliação de múltiplos planos consome uma quantidade significativa de recursos.

- **3. Planejamento Auxiliado por Planner Externo:** Esta abordagem híbrida combina a flexibilidade semântica dos LLMs com a robustez e a confiabilidade de planejadores especializados. O LLM atua como um "tradutor", convertendo o problema descrito em linguagem natural para uma representação formal (como PDDL). Em seguida, um solver simbólico ou um planejador neural, mais eficiente, gera um plano ótimo. Essa sinergia aproveita o melhor dos dois mundos: a compreensão contextual do LLM e a precisão formal dos planejadores clássicos.

- **4. Reflexão e Refinamento:** Inspirada na capacidade humana de aprender com os erros, esta metodologia atribui ao agente a capacidade de autoavaliação e autocorreção. Ao encontrar uma falha, o agente reflete sobre a causa do erro, gerando um feedback textual sobre sua própria performance, e utiliza essa reflexão para refinar o plano em tentativas subsequentes. O método Reflexion é um exemplo notável, demonstrando melhorias significativas na taxa de sucesso, embora com um custo computacional maior devido às etapas adicionais de raciocínio.

- **5. Planejamento Aumentado por Memória:** Para permitir um aprendizado contínuo, esta abordagem aprimora o agente com um módulo de memória. As experiências passadas (sucessos, falhas, conhecimento de domínio) são armazenadas e recuperadas para informar o planejamento de novas tarefas. A memória pode ser implementada de duas formas principais:
  - **Baseada em Recuperação (RAG):** As experiências são salvas em um banco de dados externo e recuperadas conforme a necessidade. É uma abordagem flexível e de baixo custo de atualização, mas sua eficácia depende da precisão do mecanismo de recuperação.
  - **Incorporada (Fine-tuning):** O conhecimento é integrado diretamente nos parâmetros do LLM por meio de ajuste fino. Embora potencialmente mais poderosa, essa abordagem é computacionalmente cara e dificulta a atualização ou o esquecimento seletivo de memórias.

Desafios Atuais e Direções Futuras

Apesar dos avanços notáveis, a área enfrenta desafios persistentes que definem a fronteira da pesquisa:

- **Alucinações e Viabilidade:** LLMs ainda podem gerar planos que incluem ações irracionais ou interações com objetos inexistentes. Garantir que os planos sejam não apenas sintaticamente corretos, mas também viáveis e aderentes às restrições do ambiente, continua sendo um obstáculo crucial.
- **Eficiência do Plano:** A maioria dos agentes atuais opera de forma "gananciosa" (greedy), focando na próxima ação lógica em vez de otimizar o plano como um todo. Isso pode levar a soluções corretas, mas subótimas e ineficientes.
- **Feedback Multimodal:** O mundo real é multimodal, mas os LLMs são primariamente textuais. A integração eficaz com modelos que processam informações visuais e auditivas é fundamental para a aplicação em cenários complexos e do mundo real.

A superação dos desafios atuais provavelmente virá da sinergia entre LLMs e outras áreas da IA, como o planejamento simbólico, sistemas de memória avançados e a capacidade de processamento multimodal, levando a criação de agentes autônomos mais robustos, eficientes e adaptáveis.

Anotações - LLMs Working in Harmony: A Survey on the Technological Aspects of Building Effective LLM-Based Multi Agent Systems

O artigo em questão apresenta uma síntese analítica sobre a construção de sistemas de múltiplos agentes baseados em Modelos de Linguagem de Grande Escala (LLMs). Embora os LLMs demonstrem capacidades notáveis, eles possuem limitações inerentes, como a tendência à alucinação e dificuldades com raciocínio abstrato complexo. A abordagem de múltiplos agentes surge como uma solução robusta para mitigar essas fraquezas, permitindo que agentes distintos colaborem, se especializem e resolvam problemas que excedem a capacidade de um único modelo.

A análise aprofunda-se nos quatro pilares que sustentam esses sistemas: **Arquitetura**, **Planejamento**, **Memória** e **Frameworks de Desenvolvimento**.

1. Arquiteturas de Colaboração entre Agentes

A arquitetura define o padrão de interação e orquestração entre os agentes para alcançar um objetivo comum. A escolha da arquitetura é uma decisão estratégica que impacta diretamente a eficiência e a sofisticação da colaboração. No artigo são citados os seguintes tipos de arquiteturas:

- **Abordagens de Agregação Simples:** Modelos como o **Agent Forest** utilizam uma lógica de "sabedoria da multidão", onde múltiplos agentes geram respostas de forma independente e a saída final é decidida por votação majoritária. Embora simples, essa abordagem pode aumentar os custos computacionais significativamente.
- **Modelos de Interação Direta:** A arquitetura **Conquer-and-Merge Discussion (CMD)** simula um debate humano, permitindo que os agentes construam suas contribuições com base nas dos outros em um processo interativo, aprimorando o raciocínio coletivo.
- **Estruturas Sequenciais:** Para tarefas de longo contexto que excedem os limites de um único LLM, a **Chain-of-Agents (CoA)** adota uma abordagem de "linha de montagem", onde agentes trabalhadores processam partes sequenciais da entrada e um agente gerente consolida os resultados.

Dentro desse cenário, a arquitetura **Mixture-of-Agents (MoA)** se destaca como um design altamente eficaz e sofisticado. Ela utiliza um modelo em camadas com agentes em papéis especializados: os **propositores**, responsáveis por gerar um leque diversificado de respostas, e os **agregadores**, que sintetizam criticamente essas propostas para formular uma saída final coesa e de alta qualidade. A principal vantagem estratégica da MoA é a capacidade de alavancar as forças de diferentes LLMs, designando-os para os papéis onde performam melhor, o que tem demonstrado resultados superiores em benchmarks de avaliação.

2. Estratégias de Planejamento e Autonomia

O planejamento confere aos agentes a capacidade de agir de forma autônoma, raciocinando sobre seus objetivos e adaptando suas ações com base no feedback do ambiente. Um planejamento robusto é a chave para mitigar alucinações e garantir que as ações sejam lógicas e eficazes.

- **Integração de Raciocínio e Ação:** O **ReAct (Reasoning and Acting)** é considerado uma abordagem fundamental. Ele intercala a geração de "traços de raciocínio" (o pensamento por trás da ação) com a execução de ações concretas. Essa sinergia permite que o agente se mantenha "aterrado", valide informações e corrija seu curso, aumentando a confiabilidade e a interpretabilidade.
- **Planejamento Adaptativo:** O **AdaPlanner** foca na flexibilidade em ambientes dinâmicos. Ele implementa um sistema de circuito fechado que permite ao agente refinar planos em tempo real, seja modificando um plano existente (refinamento no plano) ou gerando novas ações para lidar com imprevistos (refinamento fora do plano).
- **Mitigação de Alucinações de Planejamento:** O **KnowAgent** aborda diretamente a tendência dos LLMs de gerar ações incoerentes. Ele integra uma base de conhecimento de ações estruturadas que guia o processo de planejamento, garantindo que as sequências de ações geradas sejam mais razoáveis e executáveis.
- **Exploração do Espaço de Soluções:** Frameworks como o **Tree of Thoughts (ToT)** expandem o raciocínio linear, permitindo que o agente explore múltiplos caminhos de pensamento em paralelo, como os galhos de uma árvore. Essa abordagem é particularmente poderosa para problemas complexos que exigem busca e planejamento estratégico.

3. Mecanismos de Memória para Retenção de Contexto

A memória é o que permite aos agentes aprender com experiências passadas e manter o contexto ao longo de interações prolongadas. A eficácia de um agente está diretamente ligada à sua capacidade de armazenar e recuperar informações relevantes. A escolha da arquitetura de memória é crucial e depende do caso de uso:

- **Memória de Curto Prazo e Acesso a Conhecimento Externo:** A principal técnica é o **Retrieval-Augmented Generation (RAG)**, frequentemente implementada com **Bancos de Dados Vetoriais (VecDBs)**. Essa abordagem permite que o agente consulte dinamicamente uma base de conhecimento externa, aumentando seu contexto com informações atualizadas e factuais, o que é essencial para mitigar o conhecimento desatualizado do LLM.
- **Memória de Longo Prazo e Aprendizado Contínuo:** Para que os agentes evoluam com o tempo, são necessários mecanismos de memória de longo prazo. O **MemoryBank** é um exemplo notável, projetado para armazenar e atualizar memórias dinamicamente, utilizando um sistema inspirado na Curva de Esquecimento de Ebbinghaus para reforçar ou descartar memórias com base na relevância e no tempo.
- **Memória Simbólica para Precisão:** Em tarefas que exigem alta precisão e raciocínio lógico, a memória neural pode propagar erros. O **ChatDB** contorna esse problema ao integrar memória simbólica por meio de bancos de dados SQL. Isso permite que o agente realize operações de memória precisas e confiáveis, ideal para raciocínio de múltiplos saltos e manipulação de dados estruturados.

4. Frameworks e Ferramentas de Desenvolvimento

A implementação prática de sistemas de múltiplos agentes é facilitada por um ecossistema crescente de frameworks. Essas ferramentas fornecem a infraestrutura para definir agentes, gerenciar sua comunicação e orquestrar fluxos de trabalho complexos. A escolha do framework geralmente depende do paradigma de desenvolvimento:

- **Paradigma Conversacional:** O **AutoGen**, da Microsoft, é um framework poderoso para criar aplicações baseadas em conversas entre múltiplos agentes. Ele permite a definição de agentes personalizáveis que podem interagir entre si, com humanos e com ferramentas externas em um fluxo de diálogo unificado.
- **Paradigma de Fluxo de Trabalho (Workflow):** Ferramentas como **CrewAI** e **MetaGPT** são projetadas para decompor tarefas complexas em fluxos de trabalho estruturados. Elas permitem a atribuição de papéis específicos aos agentes (ex: "Pesquisador", "Escritor", "Crítico") que colaboram em sequência para atingir um objetivo, seguindo Procedimentos Operacionais Padrão (SOPs).

- **Paradigma Baseado em Grafos:** O **LangGraph** estende a popular biblioteca LangChain para permitir a criação de fluxos de trabalho cíclicos e mais complexos usando uma estrutura de grafo. Isso é especialmente útil para construir agentes que precisam iterar, modificar seu estado e tomar decisões com base em ciclos de feedback, sendo fundamental para sistemas RAG avançados.

Conclusão e Desafios Futuros

A análise dos quatro pilares revela que a construção de sistemas de múltiplos agentes eficazes depende de escolhas de design sinérgicas. A arquitetura **Mixture of Agents (MoA)** e o framework de planejamento **ReAct** emergem como práticas recomendadas para a colaboração e a autonomia dos agentes, respectivamente. As escolhas de memória e de frameworks de desenvolvimento, por sua vez, devem ser estritamente alinhadas aos objetivos específicos de cada aplicação.

Os desafios futuros residem na orquestração de um número crescente de agentes, na otimização dos altos custos computacionais e no aprimoramento da eficiência da comunicação. Superar esses obstáculos será fundamental para desbloquear todo o potencial dos sistemas de múltiplos agentes e viabilizar aplicações de IA cada vez mais sofisticadas e resilientes.

APÊNDICE 3

Termo de Aceite de Entrega

Objetivo deste documento

Este documento faz parte do Processo da disciplina Residência em IA e tem como objetivo formalizar o aceite da entrega considerando o planejado e o realizado para o período.

Data da Reunião ("Gate") de aprovação: 8 de out. de 2025

Participantes da Entrega [matriculados em Residência em IA]:

ALBERTO LUCAS BORGES DE ALMEIDA TEIXEIRA

Entrega: [descrever a ENTREGA - requisitos e produtos gerados: links para textos, códigos, vídeos etc.]

Recapitulando o meu progresso na Residência:

- Área escolhida: **Agentes Inteligentes**
- Nas primeiras Semanas, realizei um estudo mais amplo, consolidando Fundamentos e Teoria de Agentes.
- Depois, avancei para surveys mais generalistas sobre LLM-based agents.
- A partir desse entendimento, selecionei artigos mais focados nos temas identificados nos surveys, esta é minha terceira Semana nessa etapa.

Durante a sexta Semana do Processo de Residência, foram realizadas as seguintes atividades:

- Ajuste do plano de leitura desta semana
  - Previsão anterior: dois artigos de escopo amplo sobre sistemas multiagentes (último Gate).
  - Já possuo a base conceitual devido a um survey que li na última Semana.
  - Redirecionamento para artigos específicos, com foco em aplicações práticas.
  - Dei prioridade a um artigo que tinha chamado muito a minha atenção, e já estava selecionado na minha planilha de próximas leituras.
  - Observação: antes de mergulhar em artigos específicos, utilizei o ChatGPT para obter uma visão abrangente das aplicações de sistemas multiagentes. O objetivo foi clarificar o campo e identificar áreas de aplicação que mais se alinham aos meus interesses.

- Estudo do Artigo: **TradingAgents: Multi-Agents LLM Financial Trading Framework**
  - Propõe um framework que simula a dinâmica colaborativa de uma firma de trading do mundo real.
  - Estrutura dividida em **equipes**:
    - **Equipe de analistas:** análise fundamental, de sentimento de notícias e técnica;
    - **Equipe de pesquisadores:** agentes com visões opostas (otimista X pessimista), que debatem para avaliar os riscos e oportunidades de um investimento;
    - **Equipe de gestão de risco:** monitora a exposição ao risco do portfólio;
    - **Gestor do fundo:** responsável por aprovar a transação final, garantindo o alinhamento com a estratégia global;
  - **Comunicação híbrida:** relatórios estruturados e linguagem natural para os debates.

- Superou todas as 5 estratégias de baseline.
- Anotações - TradingAgents: Multi-Agents LLM Financial Trading Framework

- A leitura do artigo citado acima me encantou. A partir disso, pude finalmente definir o meu tema da residência: **Análise Preditiva de Ações com Arquitetura Multiagente** .

- Começo do estudo dos **Frameworks**:
  - Atualmente existem algumas bibliotecas em Python, que são bastante conhecidas para a criação de agentes inteligentes.
  - Dentre elas as mais conhecidas são:
    - LangChain
    - LangGraph
    - CrewAI
    - Google ADK
    - Microsoft Agent Framework
  - Diante do escopo e do tempo limitado da Residência, considero importante conhecer, ao menos de forma geral, cada um desses frameworks. Essa visão panorâmica permite compreender o contexto mais amplo, mesmo sem o aprofundamento em todos eles.
  - Diante disso, o estudo aprofundado ficará centrado apenas ao framework escolhido para a etapa prática da residência.
  - O foco desta semana foi entender, de forma geral, os fundamentos e diferenças entre os frameworks, servindo como base para um aprofundamento mais detalhado na etapa seguinte.
  - LLM Agents - Frameworks

Planejamento: **[descrever o que pretende fazer para realizar a próxima ENTREGA]**

Para a próxima semana pretendo:

- Explorar o Github do artigo: **TradingAgents: Multi-Agents LLM Financial Trading Framework**
  - Entender detalhadamente como foi feito o desenvolvimento do projeto
  - Executar o projeto localmente para validar e compreender o seu funcionamento
- Identificar e analisar potenciais pontos de melhoria que possam ser explorados na etapa prática
- Avançar no estudo dos **Frameworks**

Observação: [caso precise fazer alguma observação, de qualquer "natureza"]

ACEITE DA ENTREGA:

CEDRIC LUIZ DE CARVALHO: Go!

Anotações - TradingAgents: Multi-Agents LLM Financial Trading Framework

O artigo **"TradingAgents: Multi-Agents LLM Financial Trading Framework"**, propõe um sistema de múltiplos agentes de IA para trading no mercado financeiro, inspirado na estrutura organizacional de firmas de trading do mundo real.

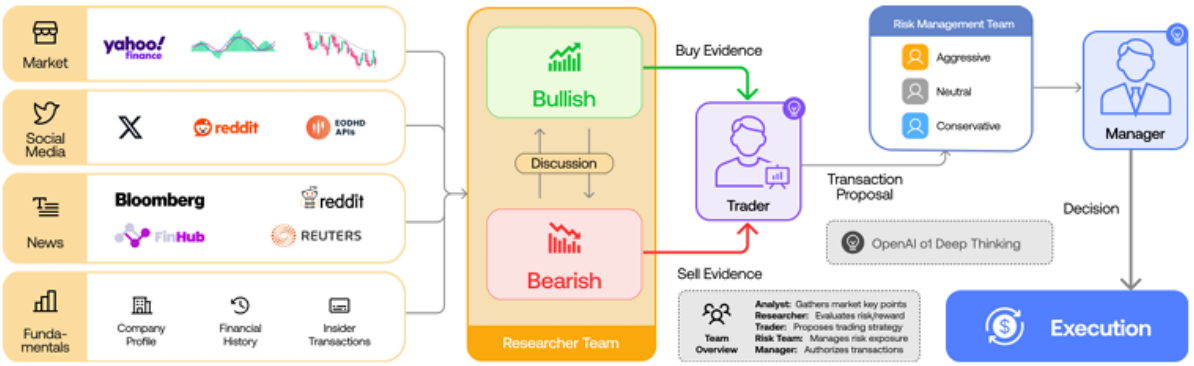
Link do repositório

1- O Problema Central e a Proposta de Valor

Modelos tradicionais de trading algorítmico e sistemas de IA de agente único falham em capturar a complexidade do mercado financeiro. O artigo identifica duas barreiras críticas que o framework **TradingAgents** visa superar:

- **Falta de Modelagem Organizacional Realista:** Sistemas existentes não replicam a dinâmica colaborativa e hierárquica das firmas de trading, focando em tarefas isoladas e ignorando fluxos de trabalho humanos eficazes na gestão de risco.
- **Comunicação Ineficiente entre Agentes:** A dependência exclusiva da linguagem natural leva ao "efeito telefone", onde a informação é corrompida e o contexto se perde em conversas longas, diminuindo a eficácia das decisões.

A proposta de valor do **TradingAgents** é uma arquitetura disruptiva que simula uma equipe de trading com especialização funcional e implementa um protocolo de comunicação híbrido para garantir integridade informacional e robustez nas decisões.



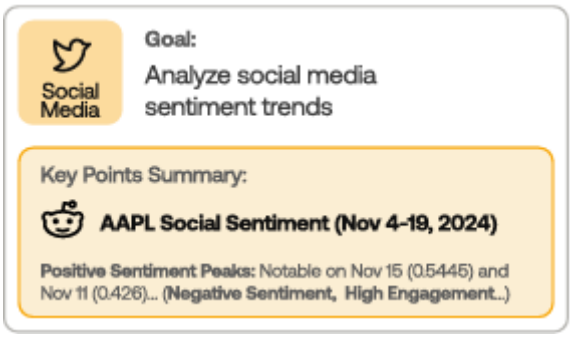
2- Arquitetura do Framework: Simulando uma Firma de Trading

Inspirado em firmas financeiras reais, o framework cria um sistema de "freios e contrapesos" com equipes de agentes especializados, operando de forma colaborativa.

Estrutura Hierárquica:

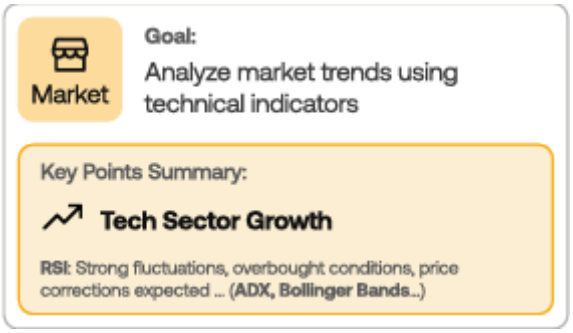
1. **Equipe de Analistas (Coleta e Processamento de Dados)**
   - **Analista Fundamental:** Avalia o valor intrínseco de empresas (demonstrações financeiras, lucros).

   - **Analista de Sentimento:** Mede o sentimento do mercado (redes sociais, notícias).



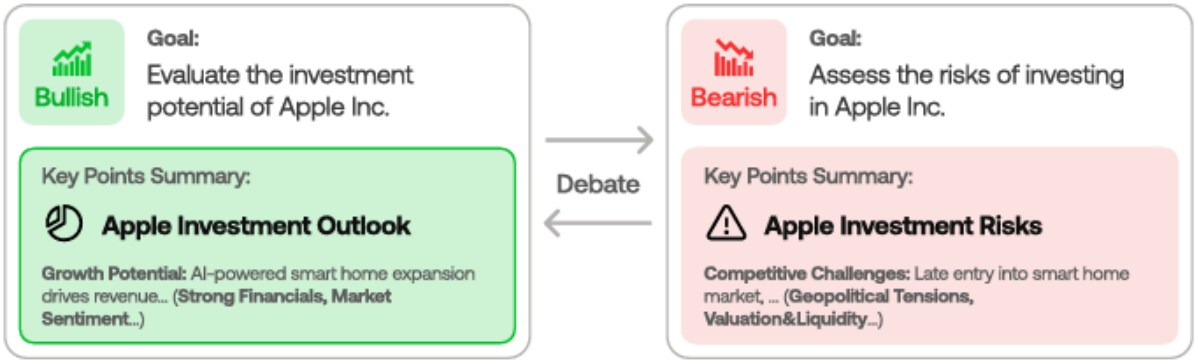
- **Analista de Notícias:** Avalia o impacto de notícias e eventos macroeconômicos.

- **Analista Técnico:** Identifica padrões de preços e volumes (indicadores como MACD, RSI).

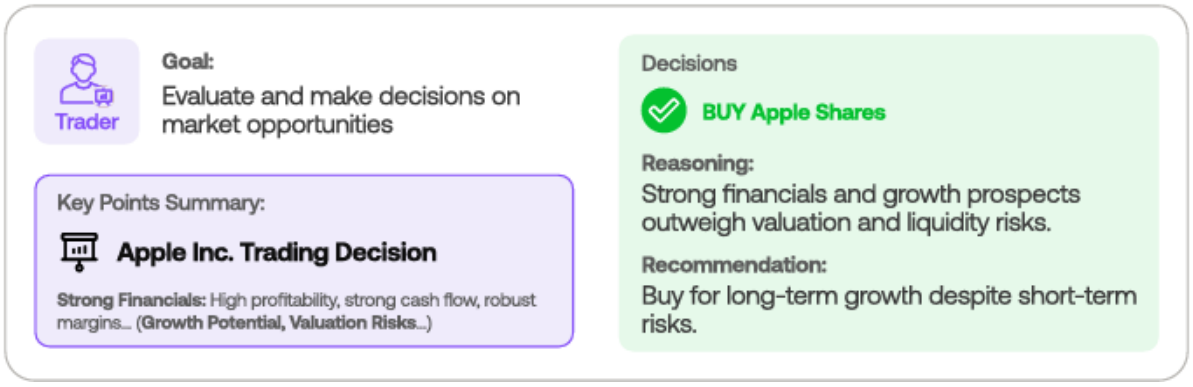


2. **Equipe de Pesquisadores (Debate e Análise Crítica)**
   - Composta por um pesquisador **Bullish** (otimista) e um **Bearish** (pessimista).

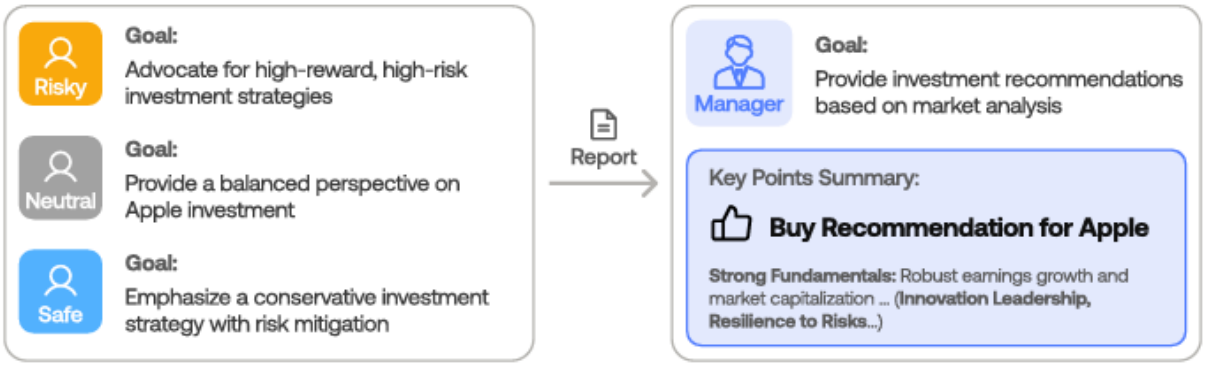
- Engajam-se em um debate dialético para construir argumentos de investimento e, simultaneamente, destacar os riscos, buscando um entendimento equilibrado do cenário.



3. **Agentes Trader (Execução)**
   - Sintetizam as análises para determinar o *timing*, o tamanho das negociações e emitir as ordens de compra/venda.



4. **Equipe de Gestão de Risco (Supervisão e Controle)**
   - Monitora a exposição ao risco (volatilidade, liquidez) e implementa estratégias de mitigação (ex: *stop-loss*).
   - Fornece feedback contínuo aos traders sobre as exposições.



5. **Gestor de Fundos (Decisão Final)**
    - Posicionado no topo da hierarquia, aprova e executa a decisão final de negociação com base nos relatórios consolidados, garantindo alinhamento com a estratégia global da firma.

3- Comunicação e Fluxo de Trabalho: O Protocolo Híbrido

O framework abandona a comunicação baseada apenas em diálogos e adota uma abordagem híbrida para garantir clareza e eficiência.

- **Comunicação Estruturada (Relatórios):** A espinha dorsal da comunicação. As equipes compilam suas análises em relatórios estruturados e concisos, armazenados em um estado global acessível. Isso evita a perda de informação ("efeito telefone") e aumenta a eficiência.
- **Diálogo Focado (Debates Estratégicos):** A linguagem natural é usada de forma direcionada para tarefas de raciocínio profundo.
    - A **Equipe de Pesquisadores** debate as teses *Bullish vs. Bearish*.
    - A **Equipe de Gestão de Risco** delibera a partir de três perspectivas (*risk-seeking*, *neutral*, *conservative*) para refinar o plano. O resultado desses debates é consolidado em relatórios para a decisão final.

4- Detalhes Técnicos da Implementação

- **Framework de Orquestração: LangGraph** é utilizado para construir e gerenciar o fluxo de trabalho complexo dos múltiplos agentes.

- **Alocação Estratégica de LLMs:**
  - **Modelos de "Pensamento Rápido" (ex: `gpt-4o-mini`):** Usados para tarefas operacionais como sumarização e coleta de dados (Analistas, Pesquisadores, Traders).
  - **Modelos de "Pensamento Profundo" (ex: `o4-mini`):** Empregados em tarefas que exigem raciocínio deliberativo e tomada de decisão crítica (Gerente de Pesquisa, Gerente de Risco).

5- Metodologia Experimental e Resultados

O desempenho do TradingAgents foi validado através de um backtesting rigoroso e comparado com cinco estratégias de mercado (baselines).

Baselines:

- **Buy and Hold (Comprar e Manter):** Consiste em investir um valor igual em todas as ações selecionadas e mantê-las durante todo o período da simulação, sem realizar vendas.
- **MACD (Convergência e Divergência de Médias Móveis):** Uma estratégia de momentum que segue tendências. Ela gera sinais de compra e venda com base nos pontos de cruzamento entre a linha MACD e a sua linha de sinal.
- **KDJ & RSI (Índice de Força Relativa):** Uma estratégia de momentum que combina os indicadores KDJ (oscilador estocástico) e RSI (índice de força relativa). O objetivo é identificar condições de "sobrecompra" (sinalizando uma possível venda) ou "sobrevenda" (sinalizando uma possível compra) de um ativo.
- **ZMR (Reversão à Média Zero):** Uma estratégia de reversão à média, que parte do princípio de que os preços dos ativos tendem a retornar à sua média. Gera sinais com base nos desvios de preço e subsequentes retornos a uma linha de referência zero.
- **SMA (Média Móvel Simples):** Uma estratégia que segue tendências e gera sinais de negociação baseados nos cruzamentos entre médias móveis de curto e longo prazo. Um sinal de compra, por exemplo, ocorre quando a média de curto prazo cruza acima da média de longo prazo.

Configuração dos experimentos:

- **Período:** 1 de janeiro de 2024 a 29 de março de 2024.
- **Ativos:** Ações de tecnologia ($AAPL, $NVDA, $MSFT, $META, $GOOGL).

- **Dados:** Multi-modais, incluindo preços, notícias, sentimento, dados financeiros e 60 indicadores técnicos.

Principais Resultados:

- **Performance Superior:** O **TradingAgents** superou todos os baselines em métricas chave. Atingiu um retorno cumulativo de **23,21%**, superando o melhor baseline em **6,1%**. Em condições adversas com $AAPL, obteve um retorno superior a **26%** em três meses.
- **Excelente Retorno Ajustado ao Risco:** O **Índice de Sharpe (SR)** foi excepcionalmente alto, indicando uma capacidade superior de gerar retornos enquanto controla o risco de forma eficaz.
- **Gestão de Risco Eficaz:** Embora algumas estratégias mais simples tenham apresentado um Rebaixamento Máximo (MDD) ligeiramente menor, foi ao custo de retornos muito inferiores. O TradingAgents manteve o MDD abaixo de **2%**, alcançando um equilíbrio ideal entre lucratividade e controle de risco.
- **Vantagem Qualitativa: Explicabilidade:** Diferente de modelos "caixa-preta", o framework fornece registros detalhados do raciocínio, debates e uso de ferramentas em linguagem natural. Essa transparência permite que operadores humanos entendam, depurem e confiem no sistema.

6- Conclusão e Principais Takeaways

O **TradingAgents** demonstra que uma arquitetura multi-agente, inspirada em estruturas organizacionais humanas, supera as limitações de sistemas de IA mais simples.

- **Especialização e Colaboração:** A integração de múltiplos agentes especializados melhora a qualidade da análise e o desempenho do trading.
- **Debate Aprimora o Raciocínio:** O confronto de perspectivas opostas (*Bullish vs. Bearish*) fortalece as decisões e a gestão de riscos.
- **Explicabilidade é uma Vantagem Competitiva:** A transparência do processo decisório é um diferencial crucial em relação aos métodos opacos de *deep learning*.
- **Adaptabilidade de Mercado:** O framework demonstrou alta capacidade de se adaptar a diferentes condições de mercado, mantendo um desempenho robusto.

Próximos Passos: Os autores planejam implantar o framework em ambiente de trading ao vivo, expandir as funções dos agentes e incorporar feeds de dados em tempo real para aprimorar ainda mais sua capacidade.

Overview - Frameworks LLM Based-Agents

1. LangChain: Orquestração de Fluxos de Trabalho Sequenciais com LLMs

Arquitetura e Filosofia: O LangChain se estabelece como um framework para a simplificação e orquestração de aplicações que interagem com LLMs. Sua filosofia central é a de "encadear" componentes em um fluxo de trabalho lógico, onde a saída de uma etapa serve como entrada para a subsequente. Ele atua como uma camada de abstração de alto nível, oferecendo uma interface padronizada para a integração de diversos LLMs, fontes de dados e ferramentas externas. Sua arquitetura é fundamentada no conceito de **Cadeia (Chain)**, que representa uma sequência de ações executadas em uma ordem predefinida. Essa estrutura linear é tecnicamente um Grafo Acíclico Dirigido (DAG), garantindo que o fluxo de dados seja unidirecional e sem loops.

Componentes Fundamentais

- **LLMs:** Módulos de integração que provêem uma interface unificada para conectar-se a diversos modelos de linguagem, como GPT-4 e Llama 2.
- **Prompts:** Templates flexíveis para a construção de instruções dinâmicas enviadas aos LLMs, permitindo a inserção de variáveis e a reutilização de contextos.
- **Chains (Cadeias):** Componente central do framework, define sequências predeterminadas de operações, como chamadas a LLMs e o uso de ferramentas. Um exemplo clássico é uma cadeia que primeiro resume um documento e, em seguida, utiliza o resumo para responder a uma pergunta.
- **Indexes (Índices):** Mecanismos para estruturar e acessar dados externos não contidos no treinamento do LLM. Incluem os *Document Loaders* (para carregar dados de fontes variadas) e os *Text Splitters* (para segmentar grandes volumes de texto).
- **Memory (Memória):** Habilita a persistência de estado entre interações, sendo essencial para a criação de assistentes conversacionais que mantêm o contexto do diálogo.
- **Agents (Agentes):** Uma evolução das cadeias, onde o LLM funciona como um motor de raciocínio. Em vez de seguir um caminho fixo, o agente avalia a entrada do usuário e decide dinamicamente qual ferramenta ou ação executar para completar a tarefa.

Aplicações e Casos de Uso Ideais

- **Tarefas Sequenciais:** Fluxos de trabalho com etapas bem definidas, como extração, sumarização e tradução de conteúdo.
- **Chatbots Simples:** Sistemas conversacionais que requerem manutenção de contexto.
- **Sistemas de Pergunta e Resposta (Q&A):** Aplicações que respondem a perguntas com base em um corpus de documentos específico.
- **Sumarização e Análise de Texto.**

2. LangGraph: Construção de Agentes Cíclicos e com Estado

Arquitetura e Filosofia: Construído sobre o LangChain, o LangGraph é uma biblioteca especializada na criação de sistemas de agentes complexos, cíclicos e com estado (stateful). Seu principal diferencial é a capacidade de modelar fluxos de trabalho não lineares, permitindo que o sistema retorne a etapas anteriores, tome decisões condicionais e modifique seu comportamento com base em interações contínuas. A estrutura linear da "cadeia" é substituída por um **Grafo (Graph)**, composto por nós e arestas.

Componentes Fundamentais

- **Nodes (Nós):** Representam unidades de trabalho, como uma função, uma chamada a um LLM ou a utilização de uma ferramenta.
- **Edges (Arestas):** Conectam os nós e governam o fluxo de execução. As arestas podem ser condicionais, permitindo que um LLM decida dinamicamente qual será o próximo nó a ser executado.
- **State (Estado):** Seu diferencial reside na implementação de um **Estado** centralizado e persistente. Este objeto é passado entre os nós, e cada nó pode ler e modificar suas informações. Isso permite a criação de loops de feedback e a manutenção de um contexto rico durante toda a execução.
- **StatefulGraph:** A estrutura de dados principal que encapsula os nós, as arestas e o objeto de estado, definindo o comportamento completo do agente.

Aplicações e Casos de Uso Ideais

- **Sistemas Multi-Agente:** Cenários onde múltiplos agentes precisam colaborar, interagir e revisar o trabalho uns dos outros de forma iterativa.
- **Assistentes Virtuais Complexos:** Aplicações que gerenciam diálogos longos e lidam com interrupções ou mudanças de objetivo pelo usuário.
- **Aplicações Interativas:** Sistemas que exigem ciclos de feedback e correção, como um agente que planeja, executa e reflete sobre uma tarefa para otimizar passos futuros.

- **Fluxos de Trabalho Não Lineares:** Processos onde a sequência de ações é imprevisível e depende de condições que evoluem dinamicamente.

3. CrewAI: Orquestração de "Equipes" de Agentes de IA

Arquitetura e Filosofia: O CrewAI é um framework de orquestração que adota uma filosofia de design que emula a dinâmica de uma equipe de especialistas humanos. A premissa é que tarefas complexas são resolvidas de forma mais eficiente através da colaboração entre múltiplos agentes autônomos, cada um com uma especialização distinta. O sistema é organizado em torno do conceito de uma **"Equipe" (Crew)**.

Componentes Fundamentais

- **Agents (Agentes):** Unidades de trabalho definidas por um papel (`role`), um objetivo (`goal`), um contexto (`backstory`) e um conjunto de `tools`. Essa configuração dota o agente de uma identidade e especialização claras.
- **Tasks (Tarefas):** Descrições atômicas de trabalho que são atribuídas a agentes específicos. As tarefas podem ter dependências entre si, formando um grafo de execução.
- **Tools (Ferramentas):** Funções e APIs externas que permitem aos agentes interagir com fontes de dados e serviços do mundo real (ex: busca na web, acesso a bancos de dados).
- **Crew (Equipe):** A composição dos agentes e tarefas. A equipe gerencia a delegação, a comunicação e a orquestração do fluxo de trabalho.
- **Process (Processo):** O método de orquestração da equipe, que pode ser sequencial (um agente passa o trabalho para o próximo) ou hierárquico (um agente "gerente" delega e valida as tarefas).

Aplicações e Casos de Uso Ideais

- **Automação de Fluxos de Trabalho Complexos:** Ideal para tarefas que mimetizam processos de negócios que requerem múltiplas competências, como planejamento de marketing, análise financeira ou desenvolvimento de software.
- **Delegação e Raciocínio Multi-etapas:** Problemas que podem ser decompostos em sub tarefas e delegados a agentes especialistas.
- **Simulação de Equipes Humanas:** Criação de sistemas de IA que espelham a colaboração, delegação e validação de uma equipe de trabalho real.

Termo de Aceite de Entrega

Objetivo deste documento

Este documento faz parte do Processo da disciplina Residência em IA e tem como objetivo formalizar o aceite da entrega considerando o planejado e o realizado para o período.

Data da Reunião ("Gate") de aprovação: 15 de out. de 2025

Participantes da Entrega [matriculados em Residência em IA]:

ALBERTO LUCAS BORGES DE ALMEIDA TEIXEIRA

Entrega: [descrever a ENTREGA - requisitos e produtos gerados: links para textos, códigos, vídeos etc.]

O meu progresso na Residência:

- Área escolhida: **Agentes Inteligentes**
- Nas primeiras Semanas, realizei um estudo mais amplo, consolidando Fundamentos e Teoria de Agentes.
- Depois, avancei para surveys mais generalistas sobre LLM-based agents.
- A partir desse entendimento, selecionei artigos mais focados nos temas identificados nos surveys.
- Tema: **Análise Preditiva de Ações com Arquitetura Multiagente.**

Durante a sétima Semana de Residência, foram realizadas as seguintes atividades:

- Análise aprofundada do artigo: **TradingAgents: Multi-Agents LLM Financial Trading Framework**
  - Reconhecendo a complexidade do tema, vi a necessidade de complementar o estudo do artigo com uma análise técnica mais profunda.
  - Iniciei, então, o estudo do código-fonte para elucidar questões e detalhes de implementação que o texto por si só não abordava.
  - Como resultado dessa análise aprofundada, elaborei um novo documento de análise do artigo, alcançando uma versão mais completa e detalhada.
  - **Trading Agents - Análise detalhada**
  - Para complementar, elaborei um quadro comparativo para mapear as funções e características de cada agente do framework.
  - **Trading Agents - Análise dos agentes**

- Estudo dos frameworks:
  - Devido ao artigo de referência ser implementado com LangGraph, priorizei o aprofundamento neste framework, pois ele será a ferramenta central na reta final do Processo de Residência.
  - Para isso comecei com um estudo introdutório, para ter uma noção geral do framework.
  - Materiais utilizados:
    - [Langgraph — Part 1 of LLM Multi-Agents series | by Tituslhy | MITB For All | Medium]

- Agentic Framework LangGraph explained in 8 minutes | Beginners Guide

- Execução do código proposto pelo artigo:
  - Análise da estrutura do código em tópicos.
  - Pontos que me chamaram a atenção:
    - **CLI Interativa:** Presença de uma CLI para monitoramento em tempo real da execução dos agentes.
    - **Flexibilidade de LLMs:** Suporte a múltiplos provedores de LLMs, incluindo a capacidade de rodar modelos localmente.
    - **Especialização de Modelos:** Permite a designação de LLMs específicos para tarefas de raciocínio rápido (quick-think) e análise profunda (deep-think).
    - **Execução Seletiva:** Modularidade que permite rodar o fluxo de análise utilizando apenas um subconjunto dos agentes analistas.
    - **Robustez na Coleta de Dados:** Implementação de um mecanismo de fallback automático entre os provedores de dados, garantindo a resiliência do sistema.
  - Identificação de oportunidades para alterações e evoluções do projeto:
    - Adaptação do projeto para a B3.
    - Implementar um módulo de avaliação (não foi disponibilizado pelos autores)
    - Análise comparativa entre a recomendação do sistema e o retorno real do ativo para calibrar o modelo de decisão.
  - **Trading Agents - Prática**

Planejamento: **[descrever o que pretende fazer para realizar a próxima ENTREGA]**

- Desenvolvimento do **Plano de Ação** para a melhoria selecionada:
  - Escopo técnico e atividades planejadas
  - Definição dos objetivos e resultados esperados
  - Plano para testes e critérios de avaliação

- Continuar com o estudo do Langgraph
  - A análise contínua do código base e o desenvolvimento de melhorias demandam um conhecimento aprofundado e constante sobre os detalhes do framework.

Observação: [caso precise fazer alguma observação, de qualquer "natureza"]

ACEITE DA ENTREGA:

CEDRIC LUIZ DE CARVALHO: Go!

Trading Agents - Análise detalhada

O Propósito do artigo:

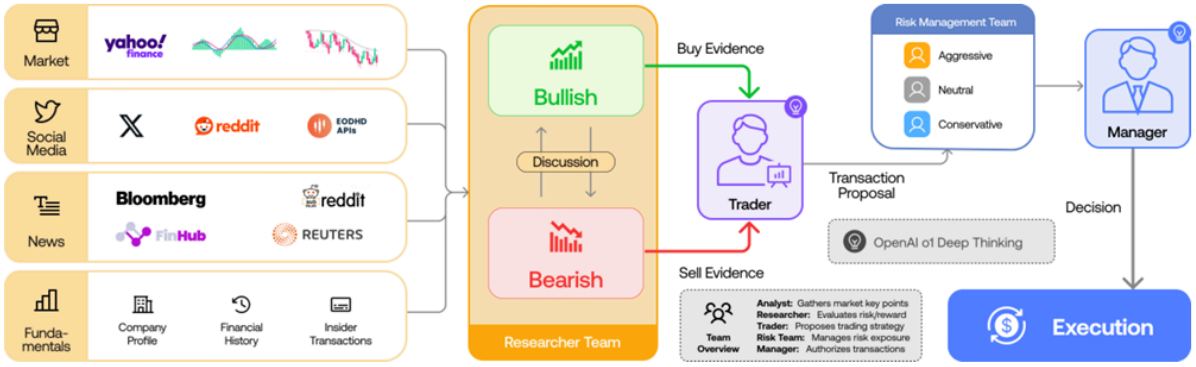
O artigo apresenta o **Trading Agents**, um framework de múltiplos agentes projetado para a tomada de decisão no mercado financeiro, que se inspira na estrutura organizacional de uma empresa de investimentos real. A sua arquitetura decompõe o complexo processo de trading em responsabilidades distintas e especializadas: um conjunto de agentes analistas focados em domínios específicos (análise técnica de mercado, notícias, sentimento social e fundamentos), que alimentam um núcleo de pesquisa. Este núcleo promove um debate adversarial (bull vs. bear) mediado por um agente "juiz" para sintetizar uma visão consolidada. A partir dessa análise, um agente trader desenvolve um plano de execução concreto, que é então submetido a um comitê de risco. Este comitê, composto por perfis com diferentes apetites ao risco (seguro, neutro, arriscado) e um gestor de risco, valida os limites operacionais e o drawdown máximo antes que a decisão final seja executada.

A principal lacuna que o **TradingAgents** busca preencher reside na fragilidade das abordagens existentes, que tendem a se situar em dois extremos ineficientes. De um lado, os pipelines baseados em um único agente, que dependem de cadeias de prompts complexas, resultando em baixa governança e pouca robustez. Do outro, sistemas multi-agentes que se baseiam em conversação livre e não estruturada, o que frequentemente leva à perda de estado, inconsistências numéricas e ambiguidades.

A solução para essa lacuna se baseia em substituir a conversação livre entre agentes por um processo formal, documentado e rigidamente orquestrado. Neste modelo, a interação é mediada por artefatos estruturados (relatórios padronizados) que servem como a única fonte factual para as deliberações. Essa abordagem permite instituir um fluxo de trabalho controlado e auditável, onde cada agente possui um papel especializado e acesso a ferramentas de dados específicas para embasar sua análise. Fundamentalmente, a governança de risco deixa de ser uma etapa final e se torna um componente intrínseco e obrigatório do fluxo, garantindo que cada plano seja validado contra critérios predefinidos.

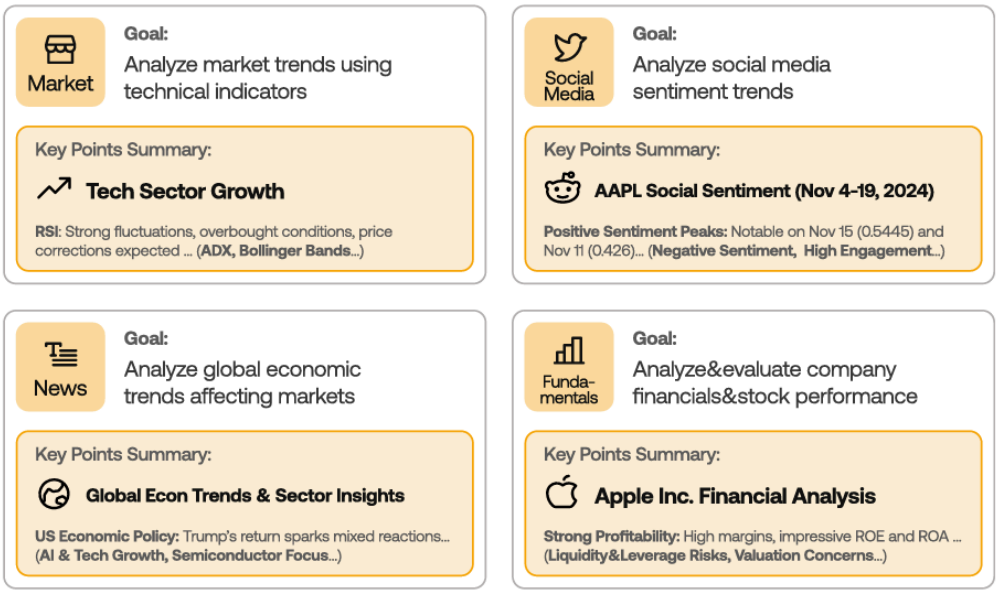
Em suma: o framework resolve a falta de coordenação, rastreabilidade e gestão de risco em agentes para trading ao **estruturar a comunicação como documentos**, **separar papéis e responsabilidades**, **ancorar tudo em dados via ferramentas** e **incluir governança de risco no próprio fluxo do sistema**. É um passo pragmático para transformar "LLM que opina" em **sistema auditável que decide**.

Visão geral da arquitetura:

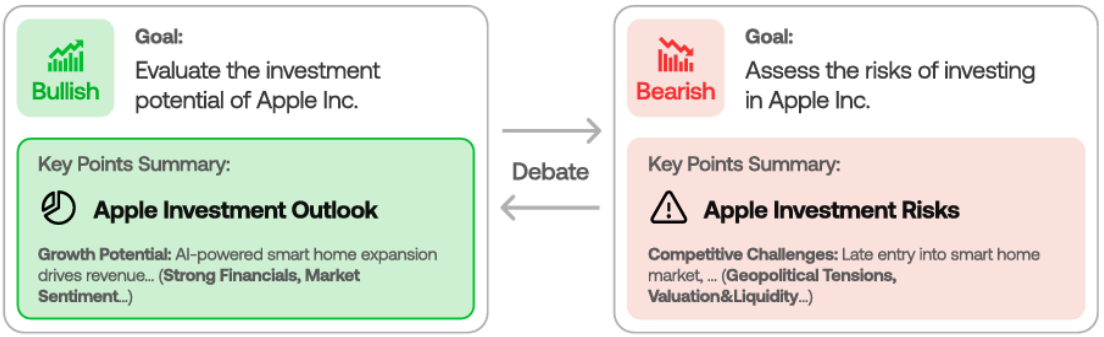


A arquitetura proposta no artigo segue 4 etapas principais:

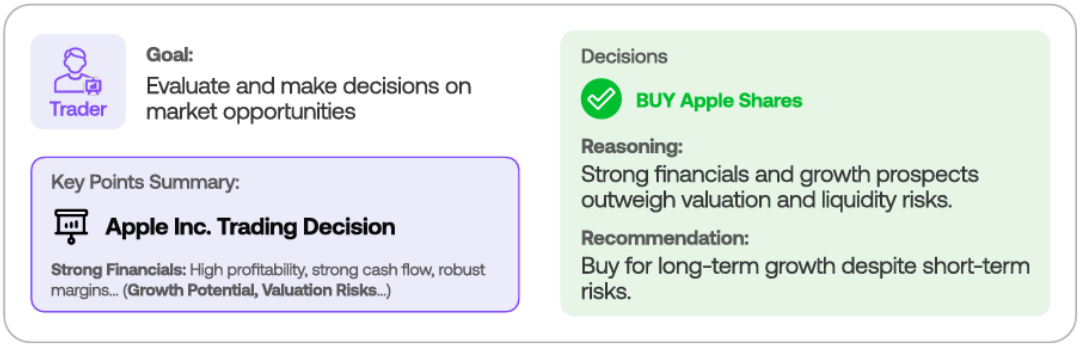
1. **Analysts:** quatro analistas trabalham em paralelo (técnico, notícias, sentimento social e fundamentos), cada um deles usando ferramentas/dados próprios e registrando suas saídas em relatórios estruturados.



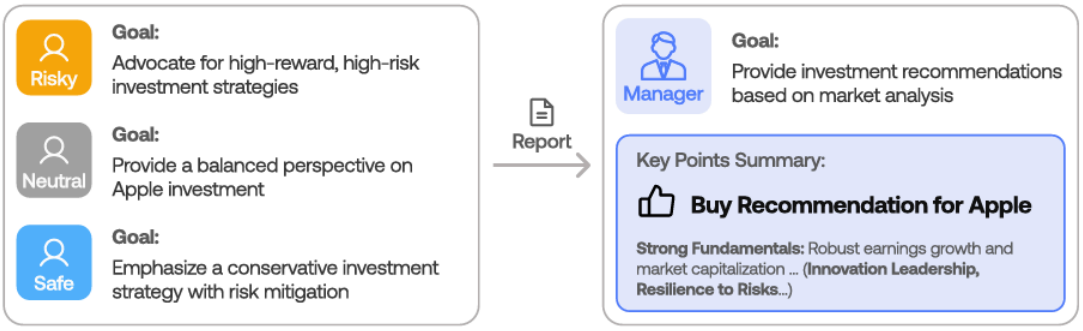
2. **Researchers (Bull vs Bear + Judge):** com base nos relatórios gerados pelos analistas, dois pesquisadores elaboram teses opostas (otimista X pessimista) e um Judge consolida os pontos fortes/fracos, reduzindo viés e alucinação.



3. **Trader:** converte a tese vencedora em um plano de trade com entradas/saídas, tamanho de posição, critérios de execução e observações operacionais.



4. **Risk Team + Fund Manager:** um conjunto de perfis de risco (convervador/neutro/arriscado) e um risk manager avaliam o plano (exposição, drawdown, aderência a limites). A aprovação do Fund Manager fecha o ciclo.



Protocolo de comunicação:

O protocolo de comunicação do **TradingAgents** foi desenhado para superar o problema do "telefone sem fio", comum em sistemas multi-agentes que dependem de conversação livre. A solução central é substituir o diálogo por um estado compartilhado e documental. A comunicação entre as etapas do processo ocorre de forma assíncrona, através de artefatos — como relatórios, históricos de debate e decisões — que são escritos no estado e servem como a base imutável para a etapa seguinte. Embora a linguagem natural seja usada para o raciocínio interno de cada agente, a comunicação inter-agentes é estritamente formalizada por meio de documentos estruturados.

Detalhamento de cada etapa da comunicação:

- **Geração de Evidências:**
  O processo é iniciado por agentes analistas dedicados a domínios específicos (mercado, notícias, sentimento e fundamentos). A responsabilidade de cada um é produzir um relatório individualizado, que servirá como alicerce para as etapas subsequentes. A geração desses relatórios é estritamente baseada em ferramentas de dados explícitas em que esses agentes têm acesso, o que garante que toda evidência seja rastreável e reprodutível, com parâmetros e fontes claramente definidos. O resultado é um conjunto de documentos factuais que formam a base da cadeia de decisão, em substituição a meras opiniões de um LLM.
- **Deliberação de Investimento:**
  Uma vez consolidados, os relatórios de análise alimentam a etapa de debate de investimento. Neste estágio, dois agentes pesquisadores com vieses opostos — otimista (*bull*) e pessimista (*bear*) — constroem teses concorrentes, registrando seus argumentos de forma estruturada. Um terceiro agente, com o papel de "juiz", avalia os relatórios iniciais e o histórico completo do debate para, então, sintetizar um "plano de investimento". Este documento finalizado torna-se a única entrada para o agente Trader, que tem a função de traduzir a tese em um plano operacional detalhado, especificando pontos de entrada, saída e o dimensionamento da posição.
- **Governança de Risco:**
  Antes de qualquer execução, o plano operacional é submetido a um rigoroso debate de risco. Nele, três agentes com perfis distintos (conservador, neutro e agressivo) analisam o plano e registram suas perspectivas. O fluxo do debate é rigidamente controlado por um marcador de "último orador" no grafo, uma regra que garante que todas as visões sejam consideradas de forma ordenada e sequencial. Ao final, um gestor de risco atua como mediador final, sendo responsável por aprovar ou rejeitar a operação e registrar a decisão em um artefato que será consumido pelo gestor do fundo.
- **Otimização e Aprendizado:**
  O ciclo de otimização e aprendizado do **TradingAgents** opera em duas frentes principais: a gestão de estado para eficiência e um mecanismo de reflexão para melhoria contínua. Para evitar a deriva de contexto e otimizar custos, o grafo aplica um "reset" após cada etapa, descartando as mensagens transitórias entre os agentes e preservando apenas os artefatos documentais — como os relatórios de análise e os registros de debate — no estado global. Em paralelo, após a conclusão de cada operação, um módulo

de reflexão é acionado. Este módulo analisa o episódio completo para gerar lições sobre o que funcionou ou falhou, criando recomendações específicas para cada um dos cinco papéis decisores (pesquisadores bull/bear, juiz de pesquisa, trader e juiz de risco). Essas lições são armazenadas em uma memória vetorial, onde a "situação" de mercado (definida pela concatenação dos quatro relatórios de análise) é indexada. Em episódios futuros, antes de tomar uma decisão, cada agente decisor consulta essa memória para recuperar as recomendações mais relevantes de situações passadas similares, injetando-as diretamente em seu prompt de raciocínio. Este mecanismo cria um ciclo de aprendizado eficiente que não requer *fine-tuning*, permitindo que o sistema reutilize experiências contextuais para reduzir a alucinação estratégica, melhorar a consistência e estabilizar o risco ao longo do tempo.

Detalhamento de cada agente:

A planilha a seguir tem como objetivo fornecer uma análise comparativa de agentes do framework, detalhando e contrastando as **ferramentas** utilizadas, seus **objetivos** específicos e o tipo de **modelo** utilizado em cada um deles.

Trading Agents - Análise dos agentes

Experimento feito no artigo:

O experimento avalia o **TradingAgents** em um cenário de teste curto, porém rico em fontes de dados, comparando-o com baselines conhecidas. A proposta é medir se a "firma virtual" de agentes gera **retorno superior e melhor ajuste risco-retorno** do que estratégias tradicionais.

- **Empresas avaliadas:** AAPL, GOOGL, AMNZ,
  - Obs: é citado um conjunto maior, mas as tabelas e figuras principais reportam resultados completos para essas três ações.
- **Período de teste:** 01-01-2024 a 29-03-2024 (diário)
  - Feito de forma que os agentes só conseguem visualizar os dados até o dia corrente.

- **Modelos empregados:**
  - Quick-thinking: gpt-4o
  - Deep-thinking: 01-preview
  - Obs: Ao utilizar o framework via CLI, o usuário tem a flexibilidade de configurar o ambiente de execução. É possível selecionar o **provedor** e, em seguida, escolher um **modelo específico** a partir de uma lista dinâmica. Devido ao design do framework, também é possível rodar os experimentos utilizando modelos rodando de forma local.
  - A orquestração dos agentes do framework foi feita utilizando **Langgraph.**
- **Ferramentas:**
  - **Preços/indicadores:** yahoo finance e/ou Alpha Vantage
  - **Notícias:** Alpha Vantage (opção via Google/Reddit (utilitários), e se configurado OpenAI web search)
  - **Sentimento social:** Alpha Vantage (opção via Google/Reddit (utilitários))
  - **Fundamentos:** Alpha Vantage
- **Orquestração:** Langgraph
- **Baselines comparativos:**
  - **Buy and Hold (Comprar e Manter):** Consiste em investir um valor igual em todas as ações selecionadas e mantê-las durante todo o período da simulação, sem realizar vendas.
  - **MACD (Convergência e Divergência de Médias Móveis):** Uma estratégia de momentum que segue tendências. Ela gera sinais de compra e venda com base nos pontos de cruzamento entre a linha MACD e a sua linha de sinal.
  - **KDJ & RSI (Índice de Força Relativa):** Uma estratégia de momentum que combina os indicadores KDJ (oscilador estocástico) e RSI (índice de força relativa). O objetivo é identificar condições de "sobrecompra" (sinalizando uma possível venda) ou "sobrevenda" (sinalizando uma possível compra) de um ativo.
  - **ZMR (Reversão à Média Zero):** Uma estratégia de reversão à média, que parte do princípio de que os preços dos ativos tendem a retornar à sua média. Gera sinais com base nos desvios de preço e subsequentes retornos a uma linha de referência zero.
  - **SMA (Média Móvel Simples):** Uma estratégia que segue tendências e gera sinais de negociação baseados nos cruzamentos entre médias

móveis de curto e longo prazo. Um sinal de compra, por exemplo, ocorre quando a média de curto prazo cruza acima da média de longo prazo.

- **Métricas de avaliação:**
  - **CR (Retorno Acumulado):**

    Mede o retorno total gerado no período da simulação:

    $$CR = \left(\frac{V_{\text{end}} - V_{\text{start}}}{V_{\text{start}}} \right) \times 100\%$$

    onde V_{end} é o valor do portfólio no fim da simulação e V_{start} é o valor inicial.

  - **AR (Retorno Anualizado):**

    Normaliza o retorno acumulado pelo número de anos:

    $$AR = \left(\left(\frac{V_{\text{end}}}{V_{\text{start}}} \right)^{\frac{1}{N}} - 1 \right) \times 100\%$$

    onde N é o número de anos na simulação.

  - **SR (Índice de Sharpe):**

    Mede o retorno ajustado ao risco, comparando o retorno em excesso do portfólio com a sua volatilidade:

    $$SR = \frac{\bar{R} - R_f}{\sigma}$$

    onde $\bar{R}$ é o retorno médio do portfólio (na mesma periodicidade das observações), R_f é a taxa livre de risco (ex.: yield do T-Bill de 3 meses) e σ é o desvio-padrão dos retornos do portfólio.

  - **MDD (Máximo Drawdown):**

Mede a maior queda "pico-a-vale" do valor do portfólio ao longo do tempo:

$$MDD = \max_{t \in [0,T]} \left(\frac{\text{Peak}_t - \text{Trough}_t}{\text{Peak}_t} \right) \times 100\%$$

- **Resultados:**

Comparação de Desempenho entre todos os métodos utilizando as quatro métricas de avaliação. Os resultados destacados em verde representam a melhor estatística de desempenho para cada modelo. A linha "Melhoria" (Improvement) ilustra os ganhos de desempenho do TradingAgent em relação aos baselines de melhor performance.

Categories	Models	AAPL				GOOGL				AMZN			
		CR%↑	ARR%↑	SR↑	MDD%↓	CR%↑	ARR%↑	SR↑	MDD%↓	CR%↑	ARR%↑	SR↑	MDD%↓
Market	B&H	-5.23	-5.09	-1.29	11.90	7.78	8.09	1.35	13.04	17.1	17.6	3.53	3.80
Rule-based	MACD	-1.49	-1.48	-0.81	4.53	6.20	6.26	2.31	**1.22**	-	-	-	-
	KDJ&RSI	2.05	2.07	1.64	1.09	0.4	0.4	0.02	1.58	-0.77	-0.76	-2.25	1.08
	ZMR	0.57	0.57	0.17	**0.86**	-0.58	0.58	2.12	2.34	-0.77	-0.77	-2.45	**0.82**
	SMA	-3.2	-2.97	-1.72	3.67	6.23	6.43	2.12	2.34	11.01	11.6	2.22	3.97
Ours	TradingAgents	**26.62**	**30.5**	**8.21**	0.91	**24.36**	**27.58**	**6.39**	1.69	**23.21**	**24.90**	**5.60**	2.11
Improvement(%)		24.57	28.43	6.57	-	16.58	19.49	4.26	-	6.10	7.30	2.07	-

Trading Agents - Prática

Mapeamento do repositório:

1. Visão Geral

- **`README.md`**: Contém a visão geral do framework, instruções de como configurar e rodar, e as chaves de API necessárias para os serviços de dados.
- **`main.py`**: Ponto de entrada principal da aplicação. É um exemplo mínimo que instancia o grafo de agentes e inicia a execução.
- **`cli/`**: Módulo responsável pela Interface de Linha de Comando (CLI), que oferece uma maneira interativa de rodar e monitorar os agentes.
    - **`main.py`**: Entrypoint do CLI, exibe o status de cada agente em tempo real.
    - **`models.py`**: Define os modelos de dados e enums usados pela interface.

2. Pacote Principal (`tradingagents/`)

Esta é a pasta principal do projeto, contendo toda a lógica dos agentes, fluxo de dados e orquestração.

2.1. Configuração Global

- **`default_config.py`**: Arquivo central de configuração. Define parâmetros globais como modelos de LLM a serem usados, provedores de dados (vendors), pastas de trabalho e número de rodadas de debate.

2.2. Os Agentes (`agents/`)

Cada agente é especializado em uma tarefa, possuindo prompts específicos e acesso a um conjunto de ferramentas (`tool calls`).

- **Analistas (`analysts/`)**: Responsáveis por coletar dados e evidências de diversas fontes.

    - **`market_analyst.py`**: Focado em análise técnica. Coleta preços históricos e indicadores (RSI, MACD, etc.) usando as ferramentas `get_stock_data` e `get_indicators`.
    - **`news_analyst.py`**: Busca por notícias gerais sobre o mercado e a empresa de interesse, utilizando `get_news` e `get_global_news`.
    - **`social_media_analyst.py`**: Analisa o sentimento em mídias sociais e notícias focadas na empresa, combinando `get_news` com fontes de dados locais (ex: Reddit).
    - **`fundamentals_analyst.py`**: Realiza a análise fundamentalista, extraindo balanço patrimonial, fluxo de caixa e demonstrativo de resultados (`get_fundamentals`, `balance_sheet`, etc.).
- **Pesquisadores (`researchers/`)**: Debatem os pontos de vista otimista (bull) e pessimista (bear) para a tese de investimento.
    - **`bull_researcher.py`**: Argumenta a favor da compra do ativo.
    - **`bear_researcher.py`**: Argumenta contra a compra ou a favor da venda.
- **Gestão de Risco (`risk_mgmt/`)**: Agentes que debatem a tese sob diferentes perfis de risco.
    - **`conservative_debator.py`**: Avalia o cenário sob um perfil de risco conservador.
    - **`neutral_debator.py`**: Avalia sob um perfil de risco neutro.
    - **`aggresive_debator.py`**: Avalia sob um perfil de risco agressivo.
- **Gerentes (`managers/`)**: Agentes "juízes" que consolidam informações e tomam decisões.
    - **`research_manager.py`**: Analisa o debate entre *bull* e *bear* e consolida uma tese de pesquisa.
    - **`risk_manager.py`**: Valida a tese contra o debate de risco e dá o veredito final.
- **Executor (`trader/`)**:
    - **`trader.py`**: Recebe o veredito final e o traduz em um plano de trade acionável (ex: pontos de entrada, saída, stop-loss).
- **Utilitários dos Agentes (`utils/`)**: Funções e ferramentas de suporte.
    - **`agent_states.py`**: Arquivo crucial que define o "shape" do estado do grafo. Descreve todas as chaves de dados que são passadas entre os nós (agentes).
    - **`..._tools.py`**: Arquivos que definem as ferramentas (`@tool`) que os agentes podem invocar para buscar dados (ex:

`core_stock_tools.py`,`news_data_tools.py`,`technical_indicators_tools.py`, `fundamental data stocks.py`).

- **`memory.py`**: Implementa a memória de longo prazo usando o **ChromaDB**. Permite que o sistema aprenda com "situações" passadas através de embeddings (vetores de texto) gerados via OpenAI.

2.3. Provedores de Dados (`dataflows/`)

Este módulo gerencia de onde e como os dados são obtidos.

- **`interface.py`**: Contém a função **`route_to_vendor`**, um roteador inteligente que mapeia uma requisição de dados (ex: "get_news") para um provedor específico (ex: Alpha Vantage), com lógica de fallback para garantir que a informação seja obtida mesmo que um provedor falhe.
- **`alpha_vantage.py`, `y_finance.py`, `google.py`**: Implementações específicas para cada API de dados.
- **`local.py`**: Um provedor de dados que utiliza um **dataset local** salvo em disco, contendo preços históricos e dados de redes sociais (Reddit). Se o dataset não estiver salvo localmente, as funções retornam vazio ou falham.

2.4. Orquestração do Grafo (`graph/`)

Utiliza a biblioteca **LangGraph** para definir o fluxo de trabalho e a interação entre os agentes.

- **`trading_graph.py`**: Define a classe principal **`TradingAgentsGraph`**, que gerencia os LLMs, as memórias e a construção do grafo.
- **`setup.py`**: **Monta a estrutura do grafo**. Define os nós (agentes) e as arestas (conexões) que determinam a ordem de execução.
- **`conditional_logic.py`**: Implementa as **regras de negócio e condicionais** que controlam o fluxo. Decide, por exemplo, se um debate precisa de mais uma rodada ou se a análise pode prosseguir.
- **`propagation.py`**: Define o **estado inicial** do grafo, populando as informações necessárias para começar a análise (ex: `company_of_interest`, `trade_date`).
- **`reflection.py`**: Implementa um processo de "pós-mortem", onde um LLM rápido analisa a execução para extrair "lições aprendidas".

- **`signal_processing.py`**: Função final que processa o texto do veredito para extrair um sinal claro e objetivo: **BUY, SELL ou HOLD**.

Observações interessantes:

- Cada tool utilizada é mapeada para um provedor (Yahoo finance, Alpha Vantage, Google/Reddit/"local"). Se um provedor falhar, há fallback previsto, garantindo a funcionalidade do sistema.
- O estado compartilhado se mantém por meio de artefatos estruturados, ao invés de apenas uma conversa entre os agentes. Dando rastreabilidade e reprodutibilidade.
- Possibilidades via CLI:
    - O usuário pode configurar facilmente os provedores de dados, seja por categoria (ex: notícias, dados fundamentalistas) ou por ferramenta específica. Além disso, a arquitetura permite selecionar diferentes modelos de LLMs (quick-think e deep-think) e seus respectivos provedores, tudo isso sem a necessidade de modificar a lógica central do grafo de execução.
    - Possibilidade de habilitar apenas alguns analistas ao executar o framework, ou seja, o grafo se adapta com o que estiver no estado.
    - É possível alterar a profundidade do Research, ajustando a quantidade de debates.

Possibilidades de alterações/melhorias:

- Adaptação do projeto para o funcionamento na B3.
- Fazer um backtest reprodutível (não foi disponibilizado pelos autores)
- Análise comparativa entre a recomendação do sistema e o retorno real do ativo para calibrar o modelo de decisão

APÊNDICE 4

Termo de Aceite de Entrega

Objetivo deste documento

Este documento faz parte do Processo da disciplina Residência em IA e tem como objetivo formalizar o aceite da entrega considerando o planejado e o realizado para o período.

Data da Reunião ("Gate") de aprovação: 23 de out. de 2025

Participantes da Entrega [matriculados em Residência em IA]:

ALBERTO LUCAS BORGES DE ALMEIDA TEIXEIRA

Entrega: [descrever a ENTREGA - requisitos e produtos gerados: links para textos, códigos, vídeos etc.]

Durante a minha especialização em **Agentes Inteligentes**:

- Nas primeiras Semanas, realizei um estudo mais amplo, consolidando Fundamentos e Teoria de Agentes.
- Depois, avancei para surveys mais generalistas sobre LLM-based agents.
- A partir desse entendimento, selecionei artigos mais focados nos temas identificados nos surveys.
- Tema escolhido: **Análise Preditiva de Ações com Arquitetura Multiagente.**

Durante a oitava Semana de Residência, foram realizadas as seguintes atividades:

- Desenvolvimento de um **web-app** via Streamlit:
  - A interface via CLI usada originalmente no Framework é interessante, mas as informações (chamada de tools, output do agentes) não fica tão "nítida".
  - Para auxiliar na visualização de cada etapa do funcionamento do Trading Agents, optei por fazer uma versão no Streamlit para substituir a interface via CLI.
  - Considero esse passo importante, pois dependendo da alteração que eu for fazer é fundamental uma análise detalhada do funcionamento do sistema.
  - Streamlit .pdf

- Tentativa de replicação do experimento do paper:
  - **Diminuição do escopo:** ao invés de analisar as 3 ações que o artigo apresenta na tabela de resultados (e demais gráficos), optei por utilizar somente a ação da Apple (AAPL).
  - **Modelos utilizados:** o4-mini e gpt-4.1-mini.
    - No artigo é citado o o1-preview para Deep Think e o gpt-4o/4o-mini como opções de modelos a serem utilizados.
    - Mas não tem nada claro citando quais modelos específicos foram utilizados nos experimentos.
  - Simulação de 01/01/2024 a 29/03/2024: cerca de 65 dias úteis.
  - **Tempo de execução:** aproximadamente 4.5 horas.
  - **Custo de chave de api:** aproximadamente 10 dólares.

- Primeira versão de um código para montar **baselines**, **calcular métricas** e comparar com o

Trading Agents:
  - evaluation.ipynb

- Resultados:
  - Nenhum pouco parecido com os que foram apresentados no artigo.
  - Pontos de dúvida:
    - Dataset para gerar as predições (qual estratégia foi utilizada?)
    - Baselines e/ou Métricas calculados de forma "errada" (diferente do que foi feito no artigo)?

- **Plano de ação:**
  - Deixar o csv de sinais do Trading Agents o mais parecido possível com um gráfico que está no artigo (é possível identificar aproximadamente quantos sinais de BUY/HOLD/SELL). Aperfeiçoar essa parte, antes de comparar as estratégias.
    - Mudança no código que utiliza o Trading Agents no período do teste.
  - Teste de outras combinações de LLMs.
  - (Possível) Teste utilizando LLMs locais para validar a diferença de performance:
    - O Framework possibilita o uso de LLMs locais (via cli)
    - Creio que não vai ser uma tarefa muito complicada, rodar o script de geração de sinais no período do teste com esses modelos
    - Comparar desempenho
  - Soluções caso essa replicação não saia como o planejado:
    - Pensar em outra estratégia para avaliar o desempenho do Trading Agents
    - Se tudo correr bem e ainda houver tempo suficiente, migrar para a B3, e validar como essa estratégia irá performar.

Planejamento: **[descrever o que pretende fazer para realizar a próxima ENTREGA]**

Continuação da parte prática utilizando o Trading Agents:
- Seguindo a partir do **plano de ação** proposto.

Observação: [caso precise fazer alguma observação, de qualquer "natureza"]

ACEITE DA ENTREGA:

CEDRIC LUIZ DE CARVALHO: Go!

APÊNDICE 5

Termo de Aceite de Entrega

Objetivo deste documento

Este documento faz parte do Processo da disciplina Residência em IA e tem como objetivo formalizar o aceite da entrega considerando o planejado e o realizado para o período.

Data da Reunião ("Gate") de aprovação: 5 de nov. de 2025

Participantes da Entrega [matriculados em Residência em IA]:

ALBERTO LUCAS BORGES DE ALMEIDA TEIXEIRA

Entrega: [descrever a ENTREGA - requisitos e produtos gerados: links para textos, códigos, vídeos etc.]

Até agora, durante a minha especialização em **Agentes Inteligentes**:

- Nas primeiras Semanas, realizei um estudo mais amplo, consolidando Fundamentos e Teoria de Agentes.
- Depois, avancei para surveys mais generalistas sobre LLM-based agents.
- A partir desse entendimento, selecionei artigos mais focados nos temas identificados nos surveys.
- Tema escolhido: **Análise Preditiva de Ações com Arquitetura Multiagente.**
- Replicação dos resultados do artigo: **TradingAgents: Multi-Agents LLM Financial Trading Framework**

No decorrer da nona Semana do processo de Residência, foram realizadas as seguintes atividades:

- Execução do plano de ação proposto no último Gate:
    - Uso do **LangSmith** para avaliar o funcionamento do Sistema Multiagentes:
        - Nenhuma discrepância em relação ao que foi proposto no Artigo sobre o funcionamento do sistema foi encontrado.
    - Teste de **LLM locais**: via Ollama e vLLM
        - Foi confirmada a viabilidade de utilizar modelos open source localmente. Porém, o desempenho geral ficou **aquém do esperado**, especialmente nos modelos de Quick Think, que são os mais frequentemente acionados pelo sistema.
        - Quick Think: a qualidade e a formatação dos relatórios com modelos locais apresentaram uma **discrepância significativa** quando comparados aos resultados obtidos com os modelos da OpenAI via API.
        - Deep Think: foi testado o GPT-OSS 20b, e o desempenho dele foi satisfatório. Devido a arquitetura proposta pelo artigo, esse modelo de Deep Think ele toma decisão com base nos relatórios e materiais gerados pelos modelos de Quick Think.
    - Geração de CSVs com as indicações (BUY/SELL/HOLD) não refletem a realidade:
        - Possuir o CSV contendo as indicações do Trading Agents **não é suficiente** para

validar a solução.
    - A partir do CSV, é necessário obter também as indicações dos baselines comparativos, e depois calcular as métricas de cada uma das estratégias de investimentos apresentada.
    - Não há informações claras sobre como fazer isso.

  - Atrás de explicações:
    - Problemas reportados no Github
    - Discord do projeto
    - Tentativa de contato com os autores do artigo (falha)

  - **Conclusão:** embora a arquitetura proposta no artigo demonstrar ser conceitualmente interessante para a solução do problema proposto, a descrição simplificada do método de teste, somada à ausência de código específico para a avaliação, inviabiliza a validação fidedigna dos resultados apresentados no artigo e levanta sérias dúvidas sobre o desempenho real da solução em comparação com o que foi alegado pelos autores.

- Análise de um novo artigo de aplicação no mercado financeiro:
  - Agent Trading Arena: A Study on Numerical Understanding in LLM-Based Agents
  - Leitura e entendimento do que foi proposto pelos autores
    - Se propõe a resolver as falhas do backtesting estático, que ignora o impacto das negociações nos preços.
    - Cria um ambiente dinâmico, onde os agentes competem.
    - Testa a capacidade estratégica e adaptativa dos agentes, não apenas a previsão.
    - Demonstra que entradas visuais e um módulo de reflexão melhoram o desempenho dos agentes.
    - Anotações - Trading Agents Arena
  - Teste de reprodução do repositório

Planejamento: **[descrever o que pretende fazer para realizar a próxima ENTREGA]**

Definição de qual abordagem seguir:

- Proposta de um método alternativo para a validação de desempenho do Trading Agents:
  - Propor comparação e cálculo de métricas
  - Possivelmente será necessário algumas mudanças em relação ao que foi proposto

- Replicação do Agent Trading Arena
  - Replicar os experimentos
  - Trazer melhor visibilidade ao fluxo de comunicação inter-agentes
  - Propor melhorias na realização da simulação

Observação: [caso precise fazer alguma observação, de qualquer "natureza"]

Agradeço ao Thiago pela indicação do LangSmith, e ao Fazzioni pela sugestão de utilizar o vLLM para rodar modelos locais.

ACEITE DA ENTREGA:

CEDRIC LUIZ DE CARVALHO: Go!

Instituto de Informática (INF) - UFG
Alameda Palmeiras, Quadra D, Câmpus Samambaia
Fone: (62) 3521-1181 / Fax: (62) 3521-1182

Anotações - Agent Trading Arena: A Study on Numerical Understanding in LLM-Based Agents

Introdução

O presente trabalho analisa a aplicação de Modelos de Linguagem Grandes (LLMs) em tarefas de raciocínio complexo, um campo que desafia paradigmas estabelecidos e abre novas fronteiras em domínios de decisão dinâmica, como o mercado financeiro. Conforme o estudo, a avaliação de agentes autônomos neste domínio é prejudicada pelo que os autores chamam de "paradoxo do mercado ao vivo". Metodologias tradicionais, como o backtesting, mostram-se incapazes de capturar a natureza interativa e adaptativa das finanças. Isso exige o desenvolvimento de ambientes de teste que possam avaliar rigorosamente a capacidade de um agente não apenas para prever, mas para influenciar e reagir a um ecossistema dinâmico.

➔ O Desafio da Avaliação de Agentes LLM em Ambientes Financeiros

A principal limitação das abordagens de avaliação existentes, segundo o artigo, é o uso do backtesting histórico. Este método opera sob a premissa fundamentalmente falha de que as ações de negociação dos agentes não influenciam os preços de mercado. Nesse modelo estático, os agentes analisam dados passivos e tomam decisões isoladas que não geram feedback realista, tratando o mercado como um ambiente fixo e não reativo.

Essa lacuna metodológica torna o backtesting estático não apenas incompleto, mas fundamentalmente inválido para avaliar agentes cujo valor principal reside em sua capacidade de interagir e influenciar um ambiente. A verdadeira inteligência estratégica em negociações emerge da habilidade de um agente de antecipar, reagir e moldar o comportamento de outros participantes — capacidades que o backtesting é incapaz de medir.

➔ A Proposta da Arena de Negociação de Agentes: Uma Solução Dinâmica

Para superar as limitações do backtesting estático, **o estudo propõe** a Arena de Negociação de Agentes, um mercado de ações virtual de soma zero onde múltiplos agentes baseados em LLM competem diretamente. Neste ambiente, as ações dos agentes impactam diretamente a dinâmica dos preços, criando um ciclo de feedback contínuo que espelha as

condições de um mercado ao vivo e permite uma avaliação autêntica do raciocínio estratégico.

O artigo destaca três contribuições principais:

- **Framework de Simulação Dinâmico:** O projeto da Arena de Negociação de Agentes, um ambiente interativo de soma zero que supera o backtesting estático, permitindo uma avaliação realista das capacidades de raciocínio e adaptação dos LLMs.
- **Análise de Raciocínio Numérico Multimodal:** A descoberta de que LLMs apresentam limitações no raciocínio numérico a partir de dados textuais, com desempenho significativamente superior quando expostos a dados visuais estruturados.
- **Otimização Estratégica via Reflexão:** A demonstração de que a incorporação de um módulo de reflexão aprimora o raciocínio e a tomada de decisão estratégica, permitindo que os agentes aprendam com a experiência para refinar continuamente suas táticas.

A necessidade de um ambiente tão sofisticado exige uma análise detalhada de sua arquitetura e dos mecanismos que o tornam uma plataforma de avaliação superior.

Arquitetura e Mecanismos da Arena de Negociação

A construção de um ambiente de simulação que replique fielmente as principais dinâmicas de um mercado real é apresentada como um pilar estratégico para a avaliação rigorosa de agentes autônomos. A Arena de Negociação de Agentes foi projetada com componentes centrais que, juntos, criam um ecossistema complexo e adaptativo.

➔ Visão Geral da Arquitetura

A arquitetura geral da Arena de Negociação de Agentes, conforme a Figura 2 do estudo, integra três componentes principais para criar um ciclo de feedback robusto:

1. **Chat Pool:** Um canal de comunicação compartilhado que simula o fluxo de informações do mercado, onde os agentes podem postar e ler notícias, análises e opiniões, introduzindo sinais informativos, ruído e desinformação.
2. **Módulos do Agente:** O núcleo de processamento de cada participante, responsável pela análise de mercado, tomada de decisão e reflexão estratégica, onde os dados são processados e as ações são formuladas.

3. **Plataforma de Negociação:** O ambiente de mercado em si, onde as ordens são executadas. Opera como um sistema em loop fechado, no qual as ações coletivas dos agentes determinam dinamicamente os preços dos ativos.

➔ Ambiente de Mercado Virtual: Um Sistema em Loop Fechado

O conceito de sistema em loop fechado é central para a Arena, representando o que os autores chamam de mudança de paradigma em relação ao backtesting. Ao eliminar dados históricos e âncoras do mundo real, o ambiente força os agentes a desenvolver um verdadeiro raciocínio estratégico baseado puramente na dinâmica interna do mercado e no comportamento de seus oponentes, sem a "muleta" dos padrões históricos.

Essa dinâmica é viabilizada pelo mecanismo de bid-ask, que modela realisticamente a formação endógena de preços. Cada ordem enviada influencia diretamente o mercado, simulando com fidelidade conceitos como liquidez, profundidade do livro de ofertas e slippage — um atrito de mercado autêntico.

➔ Mecanismos de Incentivo e Dinâmica Competitiva

Para garantir um ambiente competitivo e dinâmico, o estudo detalha como a Arena incorpora vários mecanismos de incentivo. A tabela abaixo resume cada um deles e seu impacto estratégico.

Mecanismo	**Impacto Estratégico**
Chat Pool	Simula o fluxo de informações do mundo real, incluindo sinais verdadeiros, ruído e desinformação. Aumenta a complexidade do raciocínio, forçando os agentes a filtrar e interpretar dados qualitativos.
Dividendos e Ganhos de Capital	Recompensam os agentes por manter posições de baixo custo e por valorização de ativos. Os dividendos servem como uma âncora implícita para o valor dos ativos.

Taxa Diária sobre o Patrimônio	Cria um "custo de capital" que estabelece uma linha de base de desempenho, forçando os agentes a desenvolverem estratégias que gerem retornos alfa positivos em vez de simplesmente manterem posições.
Competição de Soma Zero	Garante que o lucro de um agente seja a perda de outro. Isso cria um ambiente onde nenhuma estratégia é permanentemente ótima, forçando a adaptação contínua e o aprendizado experiencial para superar os oponentes.

A complexidade e a natureza adversária deste ambiente criam a necessidade de um agente especificamente projetado para dominá-lo, o que, segundo o artigo, leva diretamente às escolhas de design da metodologia do ArenaTrader.

Metodologia do Agente ArenaTrader

O ArenaTrader é o agente autônomo proposto no estudo, projetado para operar eficazmente na Arena de Negociação. Sua arquitetura multimodal e seu framework de tomada de decisão em três estágios são cruciais para sua capacidade de processar informações complexas e se adaptar continuamente às dinâmicas de mercado em evolução.

➔ **Representação de Dados Multimodais: Textual vs. Visual**

O agente utiliza duas modalidades de entrada de dados para formar uma compreensão abrangente do mercado.

- **Entrada Textual:** Dados de séries temporais, como preços de fechamento e volumes, são apresentados como texto. Os experimentos do estudo revelaram limitações significativas nesta modalidade: os LLMs tendem a se fixar em valores absolutos, ignorar mudanças percentuais e padrões relacionais, e dar ênfase excessiva a tendências recentes.
- **Entrada Visual:** Para superar a "miopia numérica" dos LLMs em formato textual, a entrada visual é utilizada como uma camada de abstração que traduz dados brutos em padrões estratégicos. Visualizações como gráficos de linha e de candlestick permitem que os LLMs identifiquem tendências globais, volatilidade e padrões de

longo prazo de forma mais intuitiva, integrando detalhes locais com o contexto mais amplo do mercado.

➔ Arquitetura do Agente em Três Estágios

O fluxo de processamento do ArenaTrader foi desenhado para mimetizar o comportamento de um analista humano em três estágios sequenciais.

1. **Análise de Investimentos:** Nesta fase, o agente sintetiza múltiplos sinais do Chat Pool e dos históricos de preços (textuais e visuais) para formar uma visão geral do mercado. Com base nessa análise, ele constrói um ranking de ações para identificar os ativos mais promissores.
2. **Decisão de Ação:** Integrando os relatórios de análise com dados de mercado em tempo real, o agente decide se deve comprar, vender ou manter suas posições. Ele avalia a volatilidade, o momentum dos preços e o risco do portfólio para calcular os volumes de negociação ideais, respeitando as restrições de capital.
3. **Otimização da Estratégia via Reflexão:** Ao final de cada dia de negociação, o módulo de reflexão avalia o desempenho das estratégias utilizadas, contrastando as táticas mais eficazes com as menos eficazes. A partir dessa análise, ele gera planos de ação refinados para o dia seguinte, arquivando estratégias malsucedidas e promovendo a adaptação contínua.

A validação empírica da eficácia desta arquitetura multimodal e reflexiva requer uma configuração experimental rigorosa, que é detalhada a seguir.

Configuração Experimental e Avaliação

A validação da proposta exigiu uma configuração experimental rigorosa, descrita nesta seção, para verificar a eficácia tanto da Arena de Negociação quanto do agente ArenaTrader. A seção descreve os ambientes de simulação, os modelos de base para comparação e as métricas de avaliação de desempenho.

➔ Ambiente de Simulação e Conjuntos de Dados

O ambiente da Arena foi projetado como um sistema leve e acessível. A simulação inicializa cada agente com 100.000 unidades de capital, atribuindo-lhes profissões e horizontes de

investimento variados. Os ativos virtuais também são inicializados com preços históricos e valores de dividendos por ação.

Para validação em cenários do mundo real, foram utilizados dois conjuntos de dados:

- **NASDAQ:** Sete ações líderes (AAPL, AMZA, GOOGL, MSFT, NFLX, NVDA e TSLA) foram simuladas por dois meses (3 de setembro a 29 de outubro de 2024). Os autores destacam que este período de teste foi especificamente escolhido para terminar antes do conhecimento público do modelo GPT-4o, a fim de evitar vazamentos de dados que pudessem fornecer conhecimento prévio ao sistema.
- **CSI (China):** Sete ações representativas de diferentes setores foram simuladas por um ano (2 de janeiro a 31 de dezembro de 2024).

Ambos os conjuntos de dados incluíram registros diários de preços de abertura, fechamento, máximo, mínimo e volume.

➔ **Estratégias de Linha de Base (Baselines) para Comparação**

O desempenho do ArenaTrader foi comparado a um conjunto de estratégias de baseline:

1. **Buy & Hold:** Estratégia passiva de comprar ativos no início e mantê-los até o final.
2. **SMA (Simple Moving Average):** Estratégia baseada no cruzamento de médias móveis.
3. **ZMR (Zone-Based Mean Reversion):** Estratégia que negocia com base em desvios de preços.
4. **MACD (Moving Average Convergence Divergence):** Estratégia de momentum que utiliza o cruzamento de suas linhas.
5. **StockFormer:** Modelo híbrido que combina codificação preditiva e aprendizado por reforço.
6. **TimesNet:** Modelo que transforma séries temporais 1D em representações 2D.
7. **StockMixer:** Arquitetura leve baseada em MLP que modela a dinâmica do mercado.

➔ **Métricas de Desempenho**

O desempenho foi avaliado com base em cinco métricas quantitativas:

- **Retorno Total (TR - Total Return):** Ganho ou perda percentual total do capital.
- **Taxa de Vitória (WR - Win Rate):** Proporção de dias de negociação lucrativos.

- **Índice de Sharpe (SR - Sharpe Ratio):** Mede o retorno ajustado ao risco, com uma taxa livre de risco de 0.
- **Retorno Médio Diário (Mean):** A média aritmética dos retornos diários.
- **Volatilidade do Retorno (Std):** O desvio padrão dos retornos diários, uma medida de risco.

Análise de Resultados e Descobertas

Esta seção apresenta a análise crítica dos resultados experimentais para validar as hipóteses do estudo, focando no desempenho do ArenaTrader, na importância das entradas visuais e no impacto do mecanismo de reflexão.

➔ **Desempenho de Agentes Aprimorados por Visão na Arena**

Os dados da Tabela 3 revelam que, dentro da Arena, o agente equipado com o GPT-4o alcançou o maior Retorno Total (TR) de 47,69%, evidenciando sua robusta capacidade de raciocínio orientado por dados. Para retornos ajustados à volatilidade, o GPT-4o também liderou com um Índice de Sharpe (SR) de 6,777, seguido pelo Qwen-VL-32k (SR de 1,941), confirmando a superioridade do GPT-4o na integração de sinais visuais e numéricos em mercados dinâmicos.

➔ **Validação em Dados do Mundo Real (NASDAQ e CSI)**

De forma similar, a Tabela 4 demonstra que, ao ser testado com dados dos mercados NASDAQ e CSI, o ArenaTrader (com GPT-4o) superou consistentemente todas as estratégias de baseline.

- No conjunto de dados **NASDAQ-100**, o ArenaTrader alcançou um Índice de Sharpe de **0,348**, superando o benchmark do período (0,189).
- No conjunto de dados **CSI 300**, o agente obteve um Índice de Sharpe de **0,123**, enquanto o benchmark registrou 0,054.

A Figura 6 do estudo é usada para ilustrar como a trajetória do portfólio do agente se aproxima do ótimo, especialmente durante períodos de alta volatilidade. Os autores destacam como fundamental que estes resultados de ponta, ajustados ao risco, foram alcançados através de generalização zero-shot, sem qualquer retreinamento específico para

a tarefa — uma capacidade que os modelos tradicionais, fortemente treinados, não conseguiram igualar.

➔ Estudos de Ablação: O Impacto da Modalidade e da Reflexão

Estudos de ablação foram conduzidos para isolar e medir o impacto de componentes específicos do agente.

➔ Análise Comparativa: Entradas Visuais vs. Textuais

A Tabela 5 revela uma clara hierarquia de desempenho modal: as entradas visuais melhoram substancialmente os retornos em comparação com as textuais, e a combinação de ambas as modalidades (texto + visual) alcança o melhor desempenho geral. Isso sugere que as visualizações capturam tendências globais enquanto o texto fornece detalhes precisos.

➔ Avaliação do Mecanismo de Reflexão

O módulo de reflexão provou ser um componente crucial. Conforme o estudo, a adição de reflexão ao GPT-4o com entrada visual resultou em um **ganho relativo de 41,7% no Retorno Total** (de 33,65% para 47,70%). Uma medida mais direta do valor da reflexão é vista na configuração "Textual + Visual", onde sua inclusão aumentou o TR de 26,18% para 47,69% — uma melhoria relativa de 82%. Esse salto de desempenho é diretamente atribuído à capacidade do agente de aprender com suas trajetórias diárias, impulsionando uma rápida evolução de sua estratégia baseada na experiência.

➔ Otimização da Janela Temporal de Análise

A análise do comprimento da série temporal (Tabela 6) revelou que a combinação "Textual + Visual" com uma janela de **10 dias** ofereceu o desempenho ideal, atingindo o TR (15,99%) e o SR (0,348) mais altos. Isso sugere que uma janela de 10 dias atinge um ponto ótimo entre a captura de momentum de curto a médio prazo e a filtragem de ruído de alta frequência, evitando o overfitting que pode ocorrer com janelas temporais mais longas.

Essas descobertas coletivamente validam a arquitetura do ArenaTrader, estabelecendo a base para as conclusões do estudo.

Conclusão e Limitações

O trabalho analisado apresentou a Arena de Negociação de Agentes, um framework de simulação projetado para superar as limitações do backtesting estático e servir como uma plataforma superior para a avaliação de LLMs em tarefas de raciocínio numérico e decisão dinâmica.

➔ Síntese das Contribuições e Implicações

As principais descobertas do estudo trazem implicações significativas para a pesquisa em IA financeira:

- **Superioridade da Representação Visual:** O estudo demonstrou que os LLMs têm melhor desempenho com representações visuais de dados numéricos do que com texto simples, destacando a importância do design da entrada de dados.
- **Impacto da Reflexão Estratégica:** A pesquisa confirmou que um módulo de reflexão aprimora significativamente a tomada de decisão, resultando em estratégias mais lucrativas e estáveis.
- **Validação da Arena como Testbed:** A Arena foi validada como um ambiente de teste robusto para o avanço de agentes adaptativos capazes de operar em condições próximas às do mundo real.

Coletivamente, essas descobertas sinalizam uma mudança necessária na pesquisa de IA financeira: um afastamento dos modelos de pura previsão em direção a modelos que dominam a estratégia interativa e adaptativa.

➔ Limitações e Direções Futuras

Apesar dos resultados promissores, os autores reconhecem que o ambiente virtual é uma simplificação dos sistemas financeiros reais. A ponte entre o desempenho em simulação e a aplicação prática continua sendo um desafio significativo. Portanto, o estudo conclui que o desafio central para pesquisas futuras não é apenas aprimorar o agente, mas também validar a fidelidade comportamental do próprio ambiente de simulação.

Direções futuras promissoras incluem a integração gradual de agentes em fluxos de trabalho financeiros reais, interagindo com dados ao vivo para gerar novos insights comportamentais. A expansão do escopo das tarefas e a incorporação de modalidades mais diversas também apoiarão o desenvolvimento de frameworks de avaliação mais generalizáveis e robustos.

APÊNDICE 6

Termo de Aceite de Entrega

Objetivo deste documento

Este documento faz parte do Processo da disciplina Residência em IA e tem como objetivo formalizar o aceite da entrega considerando o planejado e o realizado para o período.

Data da Reunião ("Gate") de aprovação: 12 de nov. de 2025

Participantes da Entrega [matriculados em Residência em IA]:

ALBERTO LUCAS BORGES DE ALMEIDA TEIXEIRA

Entrega: [descrever a ENTREGA - requisitos e produtos gerados: links para textos, códigos, vídeos etc.]

Durante a minha especialização em **Agentes Inteligentes**:

- Nas primeiras Semanas, realizei um estudo mais amplo, consolidando Fundamentos e Teoria de Agentes.
- Depois, avancei para surveys mais generalistas sobre LLM-based agents.
- A partir desse entendimento, selecionei artigos mais focados nos temas identificados nos surveys.
- Análise de Frameworks para a orquestração de Agentes
- Tema escolhido: **Análise Preditiva de Ações com Arquitetura Multiagente**
- Aprofundamento e tentativa de replicação dos resultados do artigo: **TradingAgents: Multi-Agents LLM Financial Trading Framework**

Recapitulando o funcionamento do **Trading Agents**:

- **Objetivo**: simular uma firma de trading, com 4 times de especialistas
  - **Analyst Team**: quatro analistas (fundamentalista, de sentimento, de notícias e técnico) produzem relatórios detalhados sobre suas respectivas áreas de análise.
  - **Researcher Team**: dois agentes (bull e bear) desenvolvem teses opostas com base nos relatórios do Analyst Team e debatem por *n* rodadas. Um terceiro agente (Research Manager) avalia o debate e escolhe a perspectiva vencedora.
  - **Trader**: sintetiza as análises quantitativas e qualitativas para emitir sinais de decisão (buy/sell/hold), apresentando uma justificativa detalhada para cada escolha.
  - **Risk Management Team**: revisa a decisão do Trader por meio de três agentes (conservador, neutro e agressivo), que debatem por *n* rodadas. O Risk Manager analisa o debate e propõe ajustes ao plano resultante, que é então submetido ao Fund Manager para aprovação e atualização final.
- **Diferencial**: protocolo de comunicação estruturado
  - Em vez de uma conversa direta, os agentes trocam **relatórios estruturados**, garantindo a preservação e rastreabilidade das informações essenciais.
- Escolha de modelos diferentes:
  - **Quick-think**: tarefas rápidas (geração de relatórios a partir de tools, debatedores)
  - **Deep-think:** tarefas de decisão

- Nas últimas Semanas, meu foco vinha sendo replicar fielmente o que foi proposto, utilizando o código disponibilizado pelos autores
    - Durante esse processo, identifiquei **diversas inconsistências**:
        - Backtest com resultados promissores, porém sem total transparência da metodologia utilizada.
        - Look-ahead bias, o sistema acessa dados de datas posteriores à data de operação
        - Prompts excessivamente simples e enviesados, comprometendo o desempenho do sistema de multiagentes
        - Cheguei à conclusão de que, com base na versão original do código, é impossível reproduzir os resultados apresentados no artigo

No decorrer da décima Semana do processo da Residência, foram realizadas as seguintes atividades:

- Após a tentativa frustrada e com base no conhecimento adquirido sobre a área, decidi **modificar o código base** para buscar um desempenho mais consistente e realista:
    - **Eliminação do look-ahead** em todas as tools que o sistema utiliza:
        - Para um dos analistas, foi necessário recorrer a dados históricos baixados manualmente, já que as APIs de mercado financeiro utilizadas não forneciam informações fundamentalistas de períodos passados (apenas as mais recentes).
        - Algo similar com o que os autores fizeram, só que eles relatam que isso foi feito para todas as tools (ponto que me chamou a atenção, já no artigo temos somente informações rasas sobre esse dataset utilizado)
    - **Mudanças nos prompts:**
        - Os analistas apresentavam viés de interpretação, pois na versão original eles demonstraram um “palpite” sobre aquilo que estavam vendo, o que era propagado para os agentes seguintes, causando inconsistências.
        - O debate do Research Team e a avaliação do Research Manager eram fracos, resultando em análises superficiais e pouco robustas.
            - Observação: o core do sistema é o Research Team, fica evidente que ele é a peça central do framework.
    - **Replicação fiel** das condições experimentais utilizadas no artigo (informado por um dos autores no Discord do paper):
        - 01/01/2024 - 29/03/2024
        - Estratégia: realização da operação no início do pregão e encerramento ao final do mesmo dia, repetindo o processo diariamente durante o backtest.

- **Resultados obtidos:**
    - **No artigo:**
        - CR%: aproximadamente 24
        - ARR%: aproximadamente 27,5
        - SR: aproximadamente 6,7
        - MDD%: aproximadamente 4,71
    - **Em meus experimentos:**
        - CR%: entre 7-13
        - ARR%: entre 7-13 (usando a mesma lógica da tabela do artigo)
        - SR: entre 3 e 5
        - MDD%: entre 5 e 9

- **Análise dos resultados:**
    - Embora os resultados obtidos não tenham alcançado os valores reportados no paper, os

testes realizados apresentam desempenho superior aos baselines (estratégias tradicionais de investimento)
- O framework demonstrou vantagem clara em relação ao uso isolado de um único LLM atuando como trader, evidenciando que a colaboração entre agentes especializados traz ganhos reais de desempenho
- Ainda assim, ficou claro que LLM-based agents, por si só, não são suficientes para operar de forma eficiente em cenários de intraday trading.
- No documento a seguir apresento uma análise detalhada sobre os resultado obtidos
- Análise de implementação - Trading Agents

Planejamento: [descrever o que pretende fazer para realizar a próxima ENTREGA]

Transição para o TCC

Observação: [caso precise fazer alguma observação, de qualquer "natureza"]

ACEITE DA ENTREGA:

CEDRIC LUIZ DE CARVALHO: Go!

Análise de implementação

Categories	Models	AAPL				GOOGL				AMZN			
		CR%↑	ARR%↑	SR↑	MDD%↓	CR%↑	ARR%↑	SR↑	MDD%↓	CR%↑	ARR%↑	SR↑	MDD%↓
Market	B&H	-5.23	-5.09	-1.29	11.90	7.78	8.09	1.35	13.04	17.1	17.6	3.53	3.80
Rule-based	MACD	-1.49	-1.48	-0.81	4.53	6.20	6.26	2.31	**1.22**	-	-	-	-
	KDJ&RSI	2.05	2.07	1.64	1.09	0.4	0.4	0.02	1.58	-0.77	-0.76	-2.25	1.08
	ZMR	0.57	0.57	0.17	**0.86**	-0.58	0.58	2.12	2.34	-0.77	-0.77	-2.45	**0.82**
	SMA	-3.2	-2.97	-1.72	3.67	6.23	6.43	2.12	2.34	11.01	11.6	2.22	3.97
Ours	TradingAgents	**26.62**	**30.5**	**8.21**	0.91	**24.36**	**27.58**	**6.39**	1.69	**23.21**	**24.90**	**5.60**	2.11
Improvement(%)		24.57	28.43	6.57	-	16.58	19.49	4.26	-	6.10	7.30	2.07	-

(tabela de resultados do artigo)

Os testes práticos para o **Trading Agents** após as alterações foram realizados utilizando as ações AAPL e GOOGL:

- Período do tempo: 01/01/2024 - 29/03/2024 (igual ao do artigo)
- Foram feitos diversos testes para os dois ativos (combinações de prompts)
- Resultados obtidos:
    - CR%: entre 7-13
    - ARR%: entre 7-13 (usando a mesma lógica da tabela do artigo, seguindo a fórmula o resultado ficaria entre 28/70)
    - SR: entre 3 e 5
    - MDD%: entre 5 e 9

Definição dos baselines e métricas:

Baselines:

- **Buy & Hold:**
  Compra os ativos no início e mantém a posição durante todo o período da simulação, sem negociações intermediárias.
- **MACD (Moving Average Convergence Divergence):**
  Estratégia de momentum seguidora de tendência que gera sinais de compra/venda pelos cruzamentos entre a linha MACD e a linha de sinal.
- **KDJ & RSI (Relative Strength Index):**
  Combina o oscilador KDJ e o RSI para identificar condições de sobrecompra e sobrevenda, disparando sinais quando essas zonas são atingidas/abandonadas.
- **ZMR (Zero Mean Reversion):**
  Estratégia de reversão à média que gera sinais com base em desvios do preço em relação a uma linha de referência zero e nas reversões subsequentes a essa referência.
- **SMA (Simple Moving Average):**

Estratégia seguidora de tendência que emite sinais quando há cruzamentos entre médias móveis simples de curto e longo prazo.

Métricas:

- **Cumulative Return (CR):**
  Mede o retorno total gerado ao longo de todo o período da simulação (do início ao fim).
- **Annualized Return (AR):**
  Converte o retorno acumulado do período para uma taxa anual equivalente, permitindo comparação entre janelas de tamanhos diferentes.
- **Sharpe Ratio (SR):**
  Mede o retorno ajustado ao risco, comparando o excesso de retorno (acima do livre de risco) com a volatilidade dos retornos.
- **Maximum Drawdown (MDD):**
  Mede a maior queda (do pico ao vale) na curva de patrimônio durante o período.

Análise dos resultados obtidos:

Nos aproximadamente 3 meses (jan–mar/2024) de day trading intraday, o portfólio apresentou **CR** entre **7% e 13%**. Usando a mesma convenção da tabela do artigo, o **ARR** ficou também entre **7% e 13%**; pela fórmula usada na tabela do artigo, o equivalente anualizado para essa janela corresponde a aproximadamente **28%–70% a.a.** O **Sharpe Ratio (SR)** situou-se entre **3** e **5**, enquanto o **máximo drawdown (MDD)** permaneceu no intervalo de **5% a 9%**.

Em conjunto, os números indicam **bom desempenho** no trimestre: retorno positivo, eficiência risco-retorno elevada (SR alto) e perdas máximas contidas (MDD moderado), compatíveis com a proposta de operação intraday.

Comentários adicionais sobre a implementação:

- Necessidades de mais testes em outros momentos do mercado para comprovar a validade da estratégia
- A maior dificuldade do Trading Agents é generalizar bem todos os padrões de mercado que ele vê, apesar disso fica nítido que com prompts adequados ele toma decisões bem.
- Para elevar o desempenho da solução, os seguintes ponto são interessantes:
  - **Ampliar indicadores de mercado:** aumentar a quantidade e diversidade de indicadores, para enriquecer o contexto entregue aos agentes.
  - **LLMs especializados em finanças nos pontos-chave:** empregar modelos financeiros em agentes críticos do fluxo (ex: Research Manager e Risk Manager) para análises mais completas e assertivas, com menor dependência de prompts extensos, que devido a natureza complexa do

mercado financeiro podem vir a causar alucinações e erros de lógica.
    - Exemplo: FIM R1 ([2503.16252] Fin-R1: A Large Language Model for Financial Reasoning through Reinforcement Learning)
- **Acoplar métodos clássicos de séries temporais:** integrar modelos tradicionais aos módulos dos debatedores e managers, de modo que os padrões numéricos cheguem pré-analisados aos LLMs, simplificando e fortalecendo a etapa de raciocínio.
- **Teste em Swing Trade:** pela natureza prescritiva do framework, que já fornece instruções de execução na versão intraday, calibrar para janelas de múltiplos dias (Swing Trade) tende a potencializar a performance, aproveitando melhor as diretrizes de entrada/saída que o sistema gera.