Evolução Colaborativa em Sistema Multiagente Baseado em LLMs: Otimização de Triagem e Diagnóstico em uma Simulação Hospitalar

Carlos Estellita Neto[1], **Thalyson Gomes Nepomuceno da Silva**[1], **Gustavo Augusto Lima de Campos**[1]

[1]Núcleo de Computação Científica e Aplicada -
Universidade Estadual do Ceará - Fortaleza, Brasil.

carlos.estellita@aluno.uece.br, thalyson.uece@gmail.com
gustavo.campos@uece.br

Abstract. *This project presents a hospital system that aims to optimize clinical screening and diagnosis through the self-evolution of agents based on LLMs. The project presents the care cycle of a general hospital, focusing on the interaction and interdependence between the Nurse Agent and Medical Agents. The main focus is to develop the interaction of agents so that they can send feedbacks to each other to improve their performance through Retrieval-Augmented Generation (RAG), supported by two memory models to store positive and negative feedbacks.*

Resumo. *Este projeto apresenta um sistema hospitalar que visa otimizar a triagem e o diagnóstico clínico por meio da auto-evolução dos agentes baseados em LLMs. O projeto apresenta o ciclo de atendimento de um hospital geral, focando na interação e na interdependência entre o Agente Enfermeiro e os Agentes Médicos. O foco principal é desenvolver a interação dos agentes de forma que possam enviar feedbacks uns para os outros para melhorarem seus desempenhos por meio de Retrieval-Augmented Generation (RAG), sustentado por dois modelos de memória para guardarem os feedbacks positivos e negativos.*

1. Introdução

Este trabalho trata do desenvolvimento de uma simulação abrangente de um hospital geral, envolvendo diferentes agentes autônomos baseados em LLMs (Modelos de Linguagem de Larga Escala). O sistema visa modelar a atuação do **Agente Enfermeiro** e do **Agente Médico**, que são representados por diferentes especialidades (Cardiologia, Neurologia e Otorrinolaringologia).

Os pacientes serão gerados dinamicamente. Em seguida, são avaliados pelo Agente Enfermeiro, que deve interpretar os sintomas do paciente e direcioná-lo ao Agente Médico com a especialidade adequada. Dessa forma, o Agente Médico deve diagnosticar a doença do paciente com base em seus sintomas e tratá-lo, evoluindo também sua própria precisão diagnóstica.

O Agente Médico deverá repassar o *feedback* ao Agente Enfermeiro com base no acerto ou erro da especialidade escolhida por ele, e quando o paciente sai do ecossistema

hospitalar, ele passará o *feedback* ao Agente Médico sobre o acerto ou erro da doença diagnosticada, servindo assim como mecanismo de aprendizado para ambos os agentes.

O objetivo principal do projeto é demonstrar a viabilidade de um sistema multiagente baseado em LLMs, no qual os agentes aprendem de forma colaborativa e interdependente, simulando o fluxo de trabalho, a divisão de tarefas e os *feedbacks* observados em ambientes hospitalares.

2. Trabalhos Relacionados

Nesta seção, serão descritos trabalhos relacionados a este contexto, em que foram produzidas simulações multiagentes hospitalares com o uso de modelos de IA (Inteligência Artificial) generativa.

Os autores em [Li et al. 2025] apresentam uma simulação hospitalar focada em doenças respiratórias para a evolução de agentes médicos. A metodologia central, denominada MedAgent-Zero, permite que um agente de LLM aprenda sem dados rotulados por humanos. O agente evolui por meio da prática simulada, armazenando interações bem-sucedidas em uma *Medical Record Library* para consultas futuras, e refletindo sobre falhas, comparando-as com a resposta correta para gerar lições aprendidas, que são armazenadas em uma base de experiência.

O estudo de [Park et al. 2023] demonstra a viabilidade de simular comportamentos sociais críveis em ambientes virtuais por meio de agentes generativos. A arquitetura proposta combina três componentes centrais: um *memory stream*, que registra continuamente as experiências do agente em linguagem natural; um mecanismo de reflexão, que sintetiza essas memórias em *insights* de nível mais elevado de forma recursiva; e um módulo de planejamento, que traduz essas reflexões em ações concretas.

Em [Tang et al. 2024], os autores propõem uma arquitetura multiagente para o raciocínio médico. O MedAgents é composto por uma equipe de agentes LLM com papéis distintos, como agente analista, agente clínico e agente revisor, que colaboram para analisar um caso, debater e chegar a um diagnóstico.

O artigo [Kim et al. 2024] foca no método de decisão para a colaboração entre LLMs no âmbito médico. A pesquisa investiga como e quando um agente deve decidir colaborar com outro. A metodologia explora estratégias adaptativas nas quais um agente avalia sua própria incerteza ou a complexidade percebida de um caso para determinar se deve tentar uma solução individualmente ou escalar o problema, solicitando a entrada de um agente especialista ou um colega de IA, visando otimizar a precisão da decisão final.

Em [Chen et al. 2025a], abordam-se os desafios de eficiência e cognição em consultas de equipes multidisciplinares usando LLMs. A metodologia implementada propõe um *framework* multiagente, onde um agente de atenção primária realiza a triagem e recruta múltiplos agentes especialistas. Para evitar a sobrecarga de informação de longos diálogos, o sistema utiliza uma estrutura de discussão residual e agregação de consenso. O sistema é auto-evolutivo, acumulando experiência através de duas bases de conhecimento distintas: uma Correct Answer Knowledge Base (CorrectKB) e uma Chain-of-Thought Knowledge Base (ChainKB).

Por fim, os pesquisadores em [Chen et al. 2025b] propõem um *framework* de Conversa Multiagente para melhorar a capacidade diagnóstica de LLMs, especialmente para

cenários médicos complexos envolvendo doenças raras. A metodologia é inspirada nas Discussões de Equipe Multidisciplinar da prática clínica. Durante a pesquisa, essa arquitetura de múltiplos agentes, com quatro agentes médicos e um agente supervisor, demonstrou ser significativamente mais precisa na formulação de diagnósticos e na sugestão de exames em comparação com agentes LLM únicos (GPT-3.5 e GPT-4).

A análise dos trabalhos relacionados demonstra um avanço progressivo no uso de LLMs em contextos médicos de agentes únicos com memória evolutiva a arquiteturas multiagentes com colaboração por debate, escalada adaptativa e bases de conhecimento compartilhadas. Entretanto, nenhuma das abordagens apresenta um ciclo de supervisão encadeada entre agentes de papéis distintos, com aprendizado persistente em ambos os níveis por meio de memórias independentes.

Tabela 1. Comparação entre trabalhos relacionados e proposta atual.

Trabalho	Agentes	Auto-evolução	Feedback entre agentes	Múltiplas especialidades	Simulação visual
Li et al. (2025)	Multi-agentes	✅	❌	✅	✅
Park et al. (2023)	Multi-agentes	✅	✅	❌	✅
Tang et al. (2024)	Multi-agentes	❌	Parcial (debate)	✅	❌
Kim et al. (2024)	Multi-agentes	❌	Parcial (escalada)	✅	❌
Chen et al. (2025a)	Multi-agentes	✅	Parcial (consenso)	✅	❌
Chen et al. (2025b)	Multi-agentes	❌	✅ Supervisionado	✅	❌
Este trabalho	**Multi-agentes**	✅	✅ **Encadeado**	✅ **(3 esp.)**	✅

Como apresentado na Tabela 1, embora os trabalhos relacionados abordem aspectos relevantes da colaboração multiagente e da auto-evolução em contextos médicos, nenhum deles reúne simultaneamente todas as características exploradas na presente proposta.

A contribuição central deste trabalho está no mecanismo de evolução encadeada dos agentes. O Agente Médico corrige o Agente Enfermeiro quando o roteamento está errado, e o paciente valida o diagnóstico ao sair do sistema — criando um ciclo de aprendizado fechado que não foi explorado nos trabalhos anteriores.

3. Fundamentação Teórica

A base cognitiva do sistema é construída a partir de LLMs, acessados via API Gemini Flash. Os LLMs são empregados como o motor cognitivo de cada agente (enfermeiro e médico), fornecendo a capacidade de raciocínio, geração de diálogo natural e classificação. A API serve como o protocolo de comunicação necessário para acessar esta inteligência e integrá-la ao sistema de simulação.

A lógica de programação do fluxo de trabalho é gerida pelo *framework* LangGraph. O LangGraph atua como orquestrador, definindo as sequências de ações do Agente Enfermeiro e dos Agentes Médicos. Ele é especificamente desenhado para definir máquinas de estado, sendo ideal para o sistema, pois gerencia as transições de estado. O Agente de Enfermeiro envia para o Neurologista, por exemplo, e os *loops* de *feedback*, como ciclo de correção dos Agentes. O LangGraph é responsável por toda a sequência

de chamadas de APIs, pelo processamento dos dados e pelo fluxo lógico que conecta o raciocínio dos agentes.

O sistema evolui por meio do Retrieval-Augmented Generation (RAG), que insere conhecimento contextual no LLM no momento da inferência. O RAG é implementado em dois módulos de memória, que compõem a estratégia de auto-evolução dos agentes: a Memória de Reforço Positivo (MRP) e a Memória de Experiência Negativa (MEN). A MRP armazena registros de casos de sucesso (reforço positivo), enquanto a MEN armazena princípios de correção de erro (reflexões) derivados de falhas passadas (reforço negativo), similar ao [Li et al. 2025].

O LangChain é utilizado para gerenciar o pipeline de RAG, e o ChromaDB atua como um banco de dados vetorial que armazena todas as coleções de memória (MRP e MEN). O processo de RAG é viabilizado pelo modelo *text-embedding-ada-002*, que converte o texto em vetores para permitir que o sistema realize buscas por similaridade semântica, garantindo que o agente recupere a experiência mais relevante em tempo real.

A base de conhecimento inicial do RAG e a geração dos pacientes simulados são fundamentadas no *dataset* AfriMedQA-v2 [Olatunji et al. 2024], um *benchmark* de questões clínicas de múltipla escolha que cobre 32 especialidades médicas, incluindo as apresentadas neste trabalho. Cada entrada do *dataset* é composta por um enunciado clínico com sintomas do paciente, opções de resposta, resposta correta, rótulo de especialidade e justificativa. O rótulo de especialidade orienta o roteamento do Agente Enfermeiro, o enunciado clínico alimenta a geração dos pacientes simulados e a resposta correta serve como gabarito para a avaliação dos Agentes Médicos e ativação dos mecanismos de memória.

Para a visualização do hospital e do fluxo de dados, são utilizadas ferramentas de desenvolvimento de jogos com uma camada de comunicação entre o *frontend* e o *backend* cognitivo. O ambiente 2D é construído com o Tiled, que define o *layout* do hospital, e o Phaser gerencia a movimentação dos agentes e as interações no mapa, traduzindo eventos visuais em comandos cognitivos para o LangGraph. Essa sincronização é realizada via API Bridge.

4. Metodologia

A metodologia deste projeto utiliza uma arquitetura de software para modelar o fluxo de trabalho clínico e demonstrar a evolução cognitiva interdependente dos agentes.

O ambiente hospitalar é projetado com o Tiled, e a movimentação e a interação dos agentes no mapa 2D são gerenciadas pelo motor lógico Phaser. Esta camada é responsável por traduzir as decisões do *backend* em movimento visual. Enquanto isso, o fluxo de trabalho é orquestrado pelo LangGraph. Este define o sistema como uma máquina de estados, gerenciando as ramificações de decisão e os *loops* de *feedback* entre o Agente Enfermeiro e os Agentes Médicos. A comunicação entre o Phaser e o LangGraph é realizada via API Bridge, que sincroniza o estado físico do simulacro com o estado cognitivo dos agentes. O raciocínio é fornecido por LLMs acessados via API.

Em relação à modelagem cognitiva dos agentes, o conhecimento e a experiência dos agentes funcionam através de um sistema de memória de longo prazo, implementado no *vector store* ChromaDB. O *text-embedding-ada-002* serve para vetorizar o texto e re-

alizar a busca semântica, garantindo que os agentes recuperem informações relevantes em tempo real. A Memória de Reforço Positivo armazena casos clínicos que resultaram em diagnóstico correto, funcionando como reforço positivo para os agentes, enquanto a Memória de Experiência Negativa armazena as falhas, servindo como reforço negativo. O ciclo de evolução dos agentes opera em dois elos de supervisão encadeados: ao receber o paciente roteado pelo Agente Enfermeiro, o Agente Médico envia *feedback* ao Enfermeiro sobre o acerto ou erro da especialidade selecionada; ao encerrar a consulta, o próprio paciente valida o diagnóstico definido pelo Agente Médico, sinalizando acerto ou erro e disparando o armazenamento na MRP ou a geração de reflexão para a MEN, respectivamente. Na Figura 1, é possível visualizar como é modelado o sistema de memória dos Agentes Enfermeiro e Médicos [Li et al. 2025].

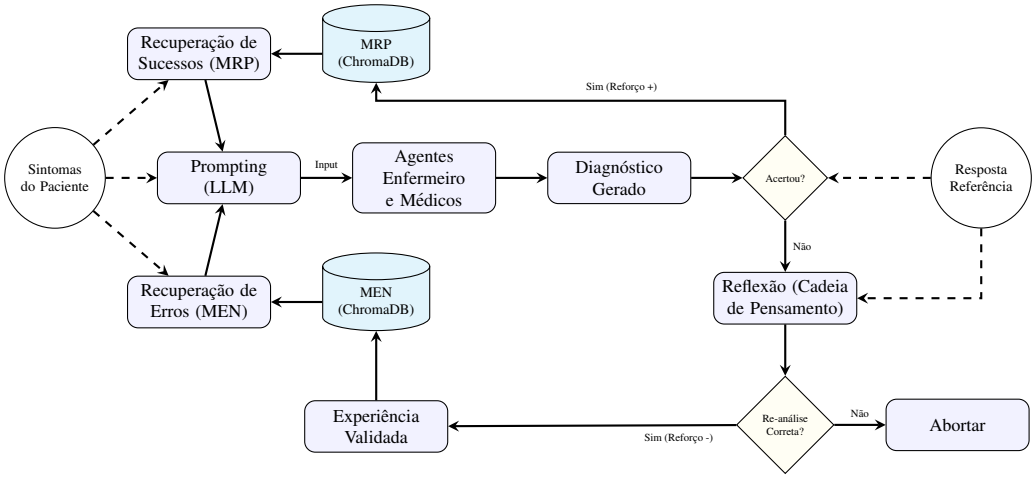

Figura 1. Fluxo de auto-evolução dos agentes Enfermeiro e Médicos.

A circulação dos pacientes pelo ambiente será da seguinte forma: Primeiramente, o paciente será atendido pelo Agente Enfermeiro na triagem, que fará o roteamento para o Agente Médico da especialidade classificada; o Agente Médico atenderá o paciente e definirá o diagnóstico. Caso interprete que se trata de outra especialidade, enviará o paciente de volta à fila de triagem e propagará o *feedback* de erro ao Agente Enfermeiro; caso contrário, o paciente sairá do hospital. Ao sair, o paciente valida o diagnóstico: se correto, o atendimento é registrado como sucesso na MRP; caso contrário, o erro é registrado na MEN e o paciente reinicia o ciclo de atendimento.

O *dataset* AfriMedQA-v2 [Olatunji et al. 2024] cumpre três papéis no sistema. Primeiro, alimenta a base de conhecimento inicial do RAG: Os casos clínicos das especialidades de cardiologia, neurologia e otorrinolaringologia são pré-processados, vetorizados pelo *text-embedding-ada-002* e armazenados no ChromaDB antes do início das simulações, fornecendo aos agentes o repertório médico por especialidade. Segundo, serve como fonte para a geração dos pacientes: As três especialidades foram escolhidas pois compartilham sintomas em comum, como vertigem e cefaleia, de forma que haja ambiguidade diagnóstica das doenças, impondo uma curva de aprendizado aos agentes. Terceiro, fornece o gabarito de avaliação: A resposta correta de cada caso determina se o roteamento do Enfermeiro e o diagnóstico do Agente Médico foram bem-sucedidos, disparando o armazenamento na MRP em caso de acerto ou a geração de reflexão para a MEN em caso de falha.

A performance será avaliada com base nas seguintes métricas para validar as hipóteses do projeto: Taxa de acerto da especialidade do Agente Enfermeiro; taxa de acerto da doença dos Agentes Médicos; Quantidade de pacientes bem tratados desde o momento em que entraram no sistema até sua saída.

5. Considerações Finais

Desta forma, o trabalho apresentado tem o intuito de desenvolver uma simulação hospitalar multiagente voltada para a otimização de fluxos clínicos por meio de agentes autônomos baseados em LLMs.

A contribuição principal deste projeto reside na implementação de um mecanismo de auto-evolução colaborativa. Diferentemente de abordagens com agentes únicos, este utiliza o ciclo de *feedback* entre o Agente Enfermeiro e os Agentes Médicos para refinar a precisão diagnóstica por meio de reforços positivos e negativos. Portanto, o trabalho não apenas simula o atendimento hospitalar, mas também o um modelo onde a inteligência coletiva dos agentes é aprimorada continuamente por meio da prática simulada e de RAG.

Para trabalhos futuros, tem-se a validação da base de conhecimento com profissionais da saúde e a expansão da tabela de doenças para cenários de maior complexidade clínica. Espera-se que esta pesquisa sirva de base para o desenvolvimento de ecossistemas digitais de saúde mais resilientes e capazes de auxiliar na tomada de decisão médica real.

Referências

Chen, K., Li, X., Yang, T., Wang, H., Dong, W., and Gao, Y. (2025a). Mdteamgpt: A self-evolving llm-based multi-agent framework for multi-disciplinary team medical consultation.

Chen, X., Yi, H., You, M., Liu, W., Wang, L., Li, H., Zhang, X., Guo, Y., Fan, L., Chen, G., Lao, Q., Fu, W., Li, K., and Li, J. (2025b). Enhancing diagnostic capability with multi-agents conversational large language models. *npj Digital Medicine*, 8(1):159.

Kim, Y., Park, C., Jeong, H., Chan, Y. S., Xu, X., McDuff, D., Lee, H., Ghassemi, M., Breazeal, C., and Park, H. W. (2024). Mdagents: An adaptive collaboration of llms for medical decision-making.

Li, J., Lai, Y., Li, W., Ren, J., Zhang, M., Kang, X., Wang, S., Li, P., Zhang, Y.-Q., Ma, W., and Liu, Y. (2025). Agent hospital: A simulacrum of hospital with evolvable medical agents.

Olatunji, T., Nimo, C., Owodunni, A., Abdullahi, T., Ayodele, E., Sanni, M., Aka, C., Omofoye, F., Yuehgoh, F., Faniran, T., et al. (2024). Afrimed-qa: A pan-african, multi-specialty, medical question-answering benchmark dataset. *arXiv preprint arXiv:2411.15640*.

Park, J. S., O'Brien, J. C., Cai, C. J., Morris, M. R., Liang, P., and Bernstein, M. S. (2023). Generative agents: Interactive simulacra of human behavior.

Tang, X., Zou, A., Zhang, Z., Li, Z., Zhao, Y., Zhang, X., Cohan, A., and Gerstein, M. (2024). Medagents: Large language models as collaborators for zero-shot medical reasoning.