Otimização de Assistentes Robóticos para Idosos: Uma Abordagem Multi-Agente e Roteamento Determinístico

Murilo Gabriel da Silva
Universidade Federal do Paraná
Curitiba, Paraná, Brasil
murilo.gabriel@ufpr.br

Luan Matheus Trindade Dalmazo
Universidade Federal do Paraná
Curitiba, Paraná, Brasil
luantrindade@ufpr.br

Eduardo Todt
Universidade Federal do Paraná
Curitiba, Paraná, Brasil
todt@ufpr.br

Abstract

This study analyzes natural interaction in assistive systems that require low latency and high reliability. Monolithic LLM models struggle in this setting because a single agent must handle reasoning and tool execution simultaneously, leading to overload and long delays. A common failure occurs when the model misses required parameters and enters repeated inference cycles, causing timeouts exceeding 300 seconds. To address this issue, a modular multi-agent architecture was designed using deterministic orchestration and intent classification. Both approaches were tested across 108 tasks per batch, including searches, weather queries, and reminders. The multi-agent system eliminated all timeout errors, increased the success rate from 37% to 98%, and reduced average latency from 204 to 26 seconds. The system also demonstrated stable and safe behavior, essential in assistive environments. Results suggest that distributing work among specialized agents with clear rules is more reliable than a single monolithic model.

Keywords

Sistemas Multi-Agente, Inteligência Artificial, LLMs, Assistente Artificial, Robô de Serviço

1 Introdução

A interseção entre inteligência artificial e gerontologia representa uma das fronteiras mais promissoras para enfrentar os desafios da transição demográfica global. O envelhecimento populacional acelerado e a consequente complexidade nos cuidados de saúde impõem uma sobrecarga crescente aos sistemas de saúde, gerando demandas que excedem a capacidade de atendimento dos recursos humanos atualmente disponíveis [1]. Nesse cenário, a inteligência artificial emerge como uma ferramenta estratégica capaz de preencher lacunas assistenciais, apoiando a prevenção e o monitoramento de condições crônicas em escalas que seriam inviáveis apenas com métodos tradicionais [3].Revisões sistemáticas da literatura indicam que, apesar de existirem desafios metodológicos e lacunas na evidência, há um corpo crescente de estudos explorando aplicações de IA em monitoramento contínuo, assistência remota, robótica e análise de dados de saúde para apoiar a prevenção, o diagnóstico, o tratamento e o acompanhamento da saúde dos idosos [4, 5]. Nesse contexto, a aplicação de Modelos de Linguagem de Grande Escala (LLMs) deixa de ser mera curiosidade tecnológica para se tornar ferramenta estrutural de suporte, com potencial para ampliar a capacidade de monitoramento, comunicação e tomada de decisão em contextos clínicos e domiciliares. Delgado e Kölling (2025) argumentam que a inserção dessas tecnologias na atenção à saúde não visa substituir o cuidado humano, mas ampliar a capacidade de monitoramento e promover a autonomia da pessoa idosa, permitindo maior independência e segurança [6].

A Estudos recentes indicam que os avanços nos LLMs representam uma mudança de paradigma em relação aos sistemas tradicionais de processamento de linguagem. Ao contrário dos modelos antigos, que dependiam de regras rígidas, os LLMs utilizam arquiteturas de Transformer[7] e treinamento em larga escala para superar essas limitações.

Essa nova abordagem permite que os modelos ofereçam respostas mais relevantes ao contexto e mantenham diálogos complexos, capturando melhor as nuances da linguagem humana.

Pesquisadores destacam que essa evolução trouxe mais do que apenas desempenho em tarefas específicas. Há uma clara transição de respostas mecânicas para interações que se aproximam do diálogo humano, tornando-se mais acolhedoras e, ao mesmo tempo, tecnicamente fundamentadas [8, 9].

1.1 Fundamentos dos Modelos de Linguagem

A base tecnológica que sustenta essa nova geração de assistentes reside nos LLMs, fundamentados na arquitetura Transformer apresentada por Vaswani et al. (2017) [7]. Antes dessa inovação, modelos de processamento de texto liam sentenças de forma linear, dificultando a compreensão de contextos longos ou ambíguos. A introdução do mecanismo de atenção permitiu que o modelo preste atenção em diferentes partes da frase simultaneamente, capturando relações de dependência complexas essenciais para entender diálogos médicos ou relatos de sintomas.

Um avanço decisivo nessa área foi o desenvolvimento do BERT (Bidirectional Encoder Representations from Transformers) [10]. Diferentemente de seus antecessores, o BERT analisa o contexto de uma palavra examinando tanto o que foi dito antes quanto o que vem depois. Essa bidirecionalidade é crítica em cenários clínicos: a diferença entre "eu tomei o remédio" e "eu devo tomar o remédio" reside em sutilezas sintáticas que modelos unidirecionais frequentemente perdem. O processo de treinamento desses modelos envolve

uma etapa de pré-treinamento massivo, onde aprendem a estrutura da língua, seguido de um fine-tuning (ajuste fino), onde são especializados para tarefas específicas, garantindo a precisão necessária para atuar em domínios sensíveis, como o de saude.

1.2 Aplicações em Saúde Cognitiva e Mental

No domínio específico da saúde mental e cognitiva, a literatura recente aponta para uma mudança de paradigma na forma como diagnóstico e monitoramento são conduzidos. Du et al. (2024) exploraram o potencial dos LLMs na análise de notas clínicas, demonstrando que esses modelos conseguem detectar padrões linguísticos sutis associados ao declínio cognitivo muito antes que os sintomas se tornem clinicamente óbvios [11]. Essa capacidade de rastreio em larga escala oferece uma janela de oportunidade vital para intervenções preventivas.[9]

Entretanto, o uso de um único modelo genérico apresenta limitações de confiabilidade. Para superar isso, Li et al. (2025) propuseram o framework CARE-AD, que representa o estado da arte na predição da Doença de Alzheimer. Em vez de confiar em uma única análise, o sistema opera como uma "junta médica virtual" composta por múltiplos agentes de IA (sistemas onde diversas unidades de software autônomas e especializadas cooperam na resolução de tarefas). Ao analisar históricos clínicos longitudinais, esses agentes colaboram entre si, mitigando o risco de alucinações (fenômeno em que a IA gera informações incorretas ou fabricadas) e oferecendo uma avaliação de risco muito mais robusta e explicável do que abordagens monolíticas (arquiteturas centralizadas dependentes de um único processador) tradicionais [12].

1.3 Sistemas Multi-Agente e Especialização

A relevância da abordagem colaborativa é evidenciada por pesquisas em sistemas multi-agente voltados ao treinamento cognitivo, conforme demonstrado por Faraziani e Eken (2025). Os autores desenvolveram uma arquitetura na qual agentes de IA adaptam dinamicamente suas estratégias segundo o desempenho do usuário [13], monitorando as respostas em tempo real para ajustar a complexidade da interação. O funcionamento baseia-se na especialização funcional que, ao contrário da centralização de processamento típica de modelos monolíticos, distribui competências entre componentes distintos. Nesta configuração, agentes operam simultaneamente em módulos dedicados, permitindo que uma unidade foque na interação conversacional enquanto outra realiza a validação estrita de dados de saúde, o que visa garantir maior robustez e precisão operacional.Nesse sentido, a arquitetura proposta transita de um modelo de inteligência centralizada para uma Orquestração de Modelos Especialistas, onde a coordenação entre agentes mimetiza um grafo de execução especializado.

1.4 Implementação de Referência: Duarte et al.

Para esta investigação, o projeto desenvolvido por Duarte et al. [2] é utilizado como referência de sistema de interação vocal baseado em grandes modelos de linguagem. A arquitetura deste sistema, ilustrada na Figura 1, opera de forma centralizada.

O fluxo de processamento inicia-se com a entrada de voz do usuário (E1), que é captada e transformada em texto utilizando motores de reconhecimento. A centralização do raciocínio ocorre no estágio (E3), onde o servidor executa a LLM Llama 3 para processar a intenção e gerar a resposta textual. Por fim, no estágio final (E4), o sistema converte a resposta e a executa em voz alta para o usuário na persona em que o sistema definiu. Este sistema funcional fundamenta-se em uma arquitetura denominada monolítica, operando mediante a atuação de um único agente de processamento central.

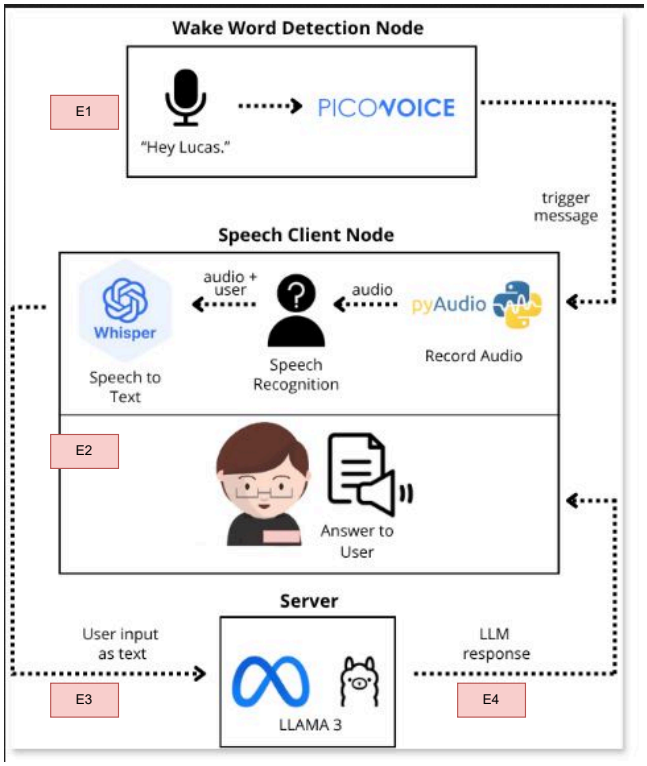

Figure 1: Fluxo da Arquitetura de Referência (Duarte et al. [2]): (E1) Detecção de palavra-gatilho via Picovoice; (E2) Transcrição de áudio com Whisper; (E3) Inferência centralizada no servidor utilizando Llama 3; (E4) Síntese de resposta para o usuário.

2 Lacunas e Motivação

Diante da necessidade de evoluir de sistemas puramente diagnósticos para assistentes executivos confiáveis, este artigo aborda a reestruturação da arquitetura de Duarte et al. [2] através de três tópicos centrais:

- Orquestração de Modelos Especialistas: Substituição do modelo monolítico unificado por uma orquestração modular baseada em grafos. Nesta abordagem, agentes especializados executam funções específicas (classificação de intenção, extração de parâmetros, validação lógica e síntese de resposta), comunicando-se através de interfaces bem definidas.
- Roteamento Determinístico: Introdução de mecanismos de validação lógica que impedem a execução de ações baseadas em inconsistências. Diferentemente de sistemas que confiam exclusivamente na capacidade do LLM de detectar erros, implementamos verificadores simbólicos que garantem a integridade antes da execução.
- Validação de Precisão Temporal: Avaliação da resposta da arquitetura distribuída frente a cenários de tempo crítico, particularmente relevante para assistência a idosos, onde a latência excessiva pode comprometer a eficácia da intervenção.

A abordagem de especialização funcional proposta alinha-se com achados recentes que demonstram que a decomposição de tarefas complexas em componentes expert melhora a performance global. A coordenação desses especialistas não apenas aumenta a acurácia, mas também gera passos intermediários de raciocínio mais claros, fator crucial para a transparência e confiança no sistema.

3 Metodologia

Para avaliar a arquitetura de múltiplos especialistas em comparação ao modelo monolítico de referência [2], este estudo utiliza uma abordagem baseada em métricas multidimensionais de desempenho.

A coleta de dados das interações foi realizada através de um processo sistematizado de geração de consultas. Para garantir a diversidade de casos de uso e o rigor nos testes, as perguntas foram elaboradas com o auxílio da LLM DeepSeek, simulando cenários complexos de interação usuário-sistema. O conjunto completo de perguntas, parâmetros de geração e dados brutos estão disponíveis para consulta nos materiais complementares (Seção 6).

3.1 Arquitetura e Tecnologias Empregadas

A arquitetura proposta adota uma abordagem híbrida e exigida em interfaces conversacionais. Diferente de sistemas puramente generativos, esta solução combina lógica determinística e probabilística para mitigar alucinações e garantir a execução correta de tarefas [8]. O ecossistema tecnológico fundamenta-se em três pilares principais:

- Classificação Semântica (BERTimbau): O sistema utiliza o BERTimbau [14] para classificação de intenções. Baseado exclusivamente em Encoders, o modelo determina a categoria semântica (ex: busca, lembrete) com baixa latência, agilizando o roteamento da decisão.
- Geração Contextual (Llama 3:8b): Para interações que exigem nuances sociais, o controle é transferido para o Llama 3:8b, responsável pela personalidade do assistente e por formular respostas empáticas [15].
- Orquestração e Segurança: A integração entre os modelos é gerida por scripts em Python e Expressões Regulares (Regex), que atuam como uma camada de limpeza e validação de fluxo [16].

O fluxo operacional, ilustrado na Figura 2, detalha como esses componentes interagem sequencialmente nos estágios de E1 a E6:

- E1 (Interface): Ponto de entrada onde o usuário envia a pergunta e recebe a resposta final.
- E2 (Limpeza): Etapa de padronização que remove caracteres estranhos e corrige o texto via Regex [16].
- E3 (Compreensão): Núcleo de classificação (BERTimbau), que identifica a intenção do usuário [14].
- E4 (Roteamento): Ponto de decisão que separa o fluxo entre busca técnica de dados ou conversação livre.
- E5 (Ferramentas): Caminho técnico que consulta APIs externas (Clima, Agenda) para informações objetivas.
- E6 (Geração): Caminho social onde o modelo Llama 3:8b cria uma resposta fluida para interações que não exigem dados externos [15].

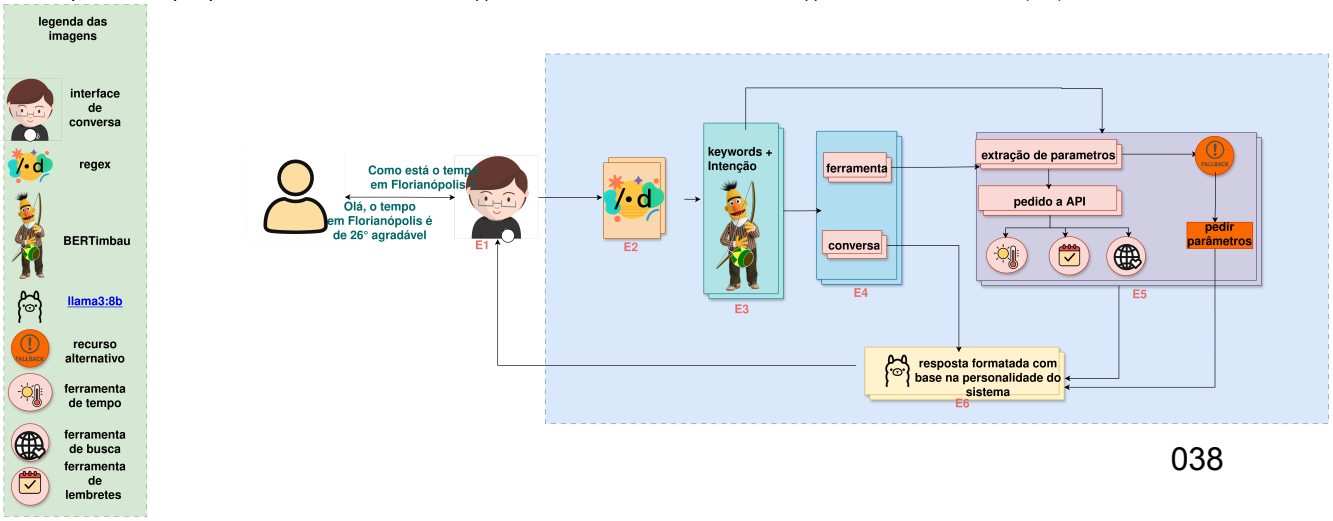

Figure 2: Fluxo de decisão da arquitetura modular mapeado por estágios: (E1) Recepção do comando; (E2) Padronização via Regex; (E3) Classificação de intenção; (E4) Roteamento condicional; (E5) Execução técnica; (E6) Geração de resposta humanizada (Llama 3:8b).
Nota: Ícones ilustrativos gerados por IA; estrutura do diagrama e fluxo lógico elaborados pelos autores.

3.2 Métricas e Corpo de Testes

A avaliação das arquiteturas foi conduzida por meio de um protocolo experimental controlado, estruturado em dois eixos principais: (i) caracterização do corpus de testes e (ii) definição de métricas de desempenho voltadas à análise temporal, operacional e de confiabilidade do sistema.

Table 1: Distribuição das categorias de teste do Lucas Assistant.

Categoria (Tipo de Pergunta)	Quantidade	Percentual (%)
Busca	10	9,26%
Clima	12	11,11%
Conversa	30	27,78%
Informação (Ruído/Casos Ambíguos)	39	36,11%
Lembrete	17	15,74%
Total	108	100,00%

O corpus experimental foi composto por 108 interações por bateria de teste distribuidas na Tabela 1, organizadas para representar situações recorrentes no cotidiano de pessoas idosas. As entradas foram classificadas segundo o tipo de intenção e o nível de estruturação linguística, incluindo desde comandos diretos até interações abertas e entradas agramaticais. Observa-se a predominância de interações de natureza conversacional e de ruído linguístico, refletindo padrões reais de uso em interfaces assistivas baseadas em linguagem natural.

As categori as relacionadas à execução de ações, como lembretes, consultas meteorológicas e buscas informativas, concentram solicitações que exigem recuperação de dados externos ou acionamento de ferramentas. Essas classes foram incluídas com o objetivo de avaliar a robustez do sistema frente a requisitos formais, como extração de parâmetros e validação lógica. Já as intenções múltiplas e condicionais foram utilizadas para testar o comportamento das arquiteturas diante de dependências semânticas explícitas e fluxos de decisão encadeados.

No que se refere às métricas de avaliação, foi adotada uma abordagem multidimensional, considerando simultaneamente eficiência computacional, estabilidade temporal e correção funcional. Esse conjunto de indicadores permite analisar o compromisso entre velocidade de resposta e confiabilidade, aspecto essencial em sistemas assistivos voltados à população idosa, que pode ser vista na Tabela 2.

Table 2: Definição das Métricas de Avaliação e Objetivos

Métrica	Definição Operacional e Objetivo
Taxa de Processamento (Throughput) [17, 18]	Def: Tokens gerados por segundo. Obj: Mensurar a latência percebida durante a geração de resposta.
Taxa de Êxito [19, 20]	Def: % de solicitações concluídas corretamente. Obj: Validar a confiabilidade e precisão funcional do sistema.
Correlação Temporal [21, 22]	Def: Coeficiente de Pearson comparando curvas de tempo. Obj: Verificar se a complexidade da tarefa afeta proporcionalmente ambas as arquiteturas.
Estabilidade Temporal [23, 24]	Def: Variação de latência/erros em sessão contínua. Obj: Detectar degradação de desempenho (“fadiga”) em interações longas.
Perfil de Latência [25, 26]	Def: Histograma dos tempos de resposta. Obj: Avaliar a previsibilidade e constância temporal, fundamentais para a experiência do usuário idoso.

Nota: Def. = Definição conforme fonte; Obj. = Objetivo de mensuração.

Taxa de Processamento foi empregada como indicador de latência percebida, quantificando a velocidade de geração de respostas ao longo das interações. A Taxa de Êxito mensurou a proporção de solicitações concluídas corretamente, sendo particularmente relevante em tarefas críticas, como lembretes de medicação e agendamentos. Métricas de caráter temporal, como correlação e estabilidade, foram utilizadas para verificar a consistência do desempenho ao longo de sessões contínuas, enquanto o perfil de latência permitiu avaliar a previsibilidade do tempo de resposta.

Esse conjunto de métricas possibilita uma análise comparativa abrangente entre arquiteturas monolíticas e modulares, evidenciando não apenas diferenças médias de desempenho, mas também padrões de degradação, variabilidade e comportamento em cenários de maior complexidade.

4 Resultados e Discussão

4.1 Distribuição de Throughput

A análise quantitativa concentrou-se no throughput, métrica que define a capacidade de processamento do sistema em tokens por segundo [17, 18]. A Figura 3 apresenta o histograma comparativo de desempenho entre as arquiteturas avaliadas.

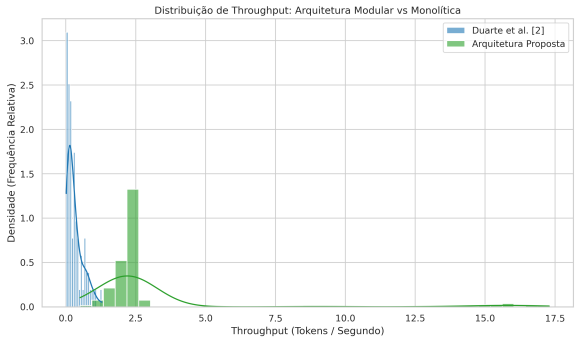

Figure 3: Distribuição de Throughput: O perfil bimodal da arquitetura proposta (verde) evidencia a alternância dinâmica de recursos, em contraste com a uniformidade do modelo monolítico (azul).

Os dados revelam padrões de execução distintos. A arquitetura monolítica manteve um ritmo de processamento constante e contido, limitado pela necessidade de processar todo o contexto em uma única inferência. Em contraste, a arquitetura modular exibiu uma distribuição bimodal, reflexo direto do mecanismo de roteamento seletivo:

- Região de Alta Vazão: Os picos à direita no gráfico correspondem ao fluxo generativo (nó Llama 3), onde o sistema responde a interações conversacionais com latência mínima, livre de validações externas.
- Região de Processamento Intensivo: A cauda à esquerda representa o fluxo determinístico, onde a velocidade de geração é naturalmente reduzida pelo tempo de latência de rede (RTT) das APIs e pelas verificações de segurança dos parâmetros.

Essa separação demonstra a eficiência computacional do modelo proposto: o sistema evita impor o custo temporal de validações lógicas complexas a interações simples, alocando recursos de processamento pesado apenas quando a intenção do usuário exige rigor factual.

4.2 Taxa de Sucesso por Tipo de Pergunta

A estratificação dos resultados por domínio funcional evidencia o impacto do roteamento híbrido na integridade das respostas [? ?]. Nas categorias que demandam dados estruturados, especificamente "Lembretes" e "Clima", a abordagem modular registrou incidência significativamente menor de alucinações quando comparada ao modelo monolítico, correlacionada à verificação determinística de parâmetros implementada antes da execução de ferramentas. O mecanismo de fallback, que solicita dados faltantes ao usuário, preveniu 100% dos erros de execução prematura observados na arquitetura de referência.

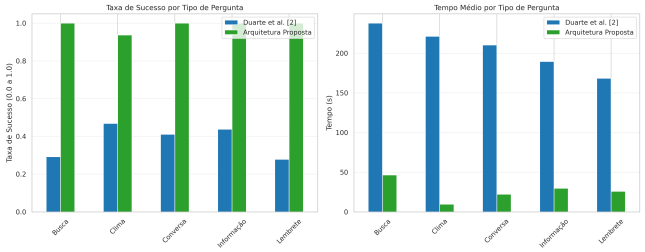

Figure 4: Desempenho Segmentado por Intenção

Por outro lado, na categoria conversa, as métricas de desempenho mantiveram-se estatisticamente equivalentes entre os dois modelos (diferença inferior a 2%), indicando que a modularização não compromete a capacidade de interação social. Este resultado valida a hipótese de que a separação de responsabilidades não prejudica a naturalidade da conversa, mantendo a empatia e fluidez necessárias para engajamento de longo prazo com usuários idosos.

4.3 Análise de Precisão na Extração de Ferramentas

A eficácia do mecanismo de extração de intenções foi avaliada comparando-se a ferramenta invocada pelo modelo com o gabarito esperado. A análise quantitativa, detalhada na Figura 5, revela uma divergência fundamental no comportamento de falha entre as arquiteturas.

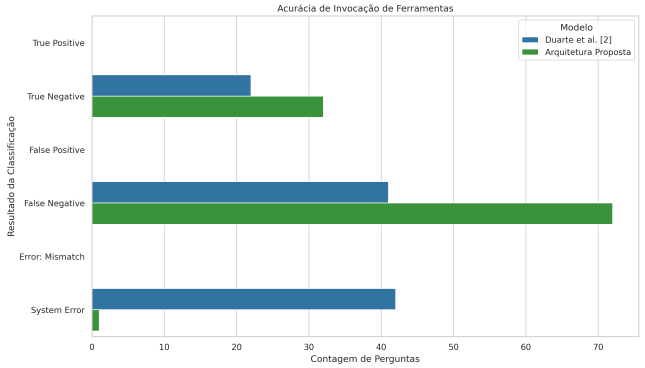

Figure 5: Classificação da Acurácia de Invocação: Comparativo entre erro sistêmico e conservadorismo na extração.

O modelo de referência (Duarte et al.)[2] demonstrou instabilidade crítica, registrando uma incidência elevada de System Errors (aproximadamente 42 ocorrências). Isso indica que a arquitetura monolítica frequentemente entra em ciclos de timeout ou falhas de conexão ao tentar processar chamadas de ferramentas. É importante ressaltar que, como ambos os testes foram conduzidos sob a mesma plataforma de execução e hardware, essa instabilidade é atribuída à sobrecarga lógica do modelo unificado, e não a limitações de infraestrutura.

Em contrapartida, a Arquitetura Proposta reduziu os erros sistêmicos a um nível negligenciável (< 2 ocorrências), validando a robustez do orquestrador modular. Contudo, observa-se uma migração das falhas para a categoria False Negative (barra verde predominante, ≈ 72 ocorrências).

Este fenômeno aponta para um comportamento "conservador" do algoritmo de extração proposto: diante de ambiguidades ou da ausência de parâmetros obrigatórios, o sistema opta por não invocar a ferramenta (não gerando a tag `[Invoca ferramenta]`), preferindo fornecer uma resposta conversacional padrão ou solicitar esclarecimentos. Embora essa estratégia reduza a taxa de automação imediata (True Positives), ela elimina o risco de execuções errôneas e travamentos, priorizando a estabilidade operacional e a integridade do sistema.

4.4 Distribuição dos Tempos de Resposta

Conforme ilustrado na Figura 6, a distribuição de latência da arquitetura proposta exibe um perfil bimodal, refletindo o fluxo de dados no grafo de execução [25, 26]. O primeiro pico de frequência (20-30s) concentra as interações sociais processadas localmente pelo LLM, enquanto a cauda da distribuição (40-60s) representa as requisições que envolvem chamadas de API. Esta dispersão dos dados quantifica a variação temporal introduzida pela dependência de respostas de redes externas.

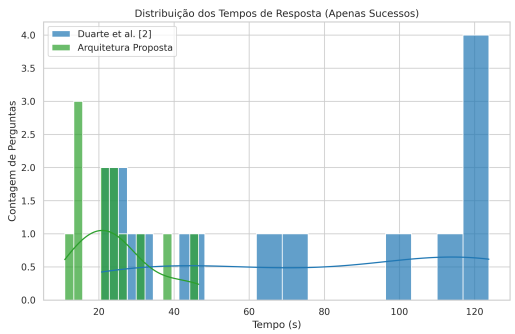

Figure 6: Histograma de Latência: Perfil bimodal da proposta vs. cauda longa da referência.

Diferentemente da métrica de desvio padrão global, que perde significado em distribuições bimodais, a análise focada nos intervalos de confiança revela que a arquitetura modular oferece maior previsibilidade local. Enquanto o modelo monolítico apresenta uma dispersão errática (cauda longa imprevisível), a proposta confina os tempos de resposta a janelas temporais bem definidas. Para usuários idosos, essa consistência é crítica: tempos de resposta que, embora variem, mantêm-se dentro de faixas esperadas, evitam a ansiedade ou a percepção de falha no sistema, reduzindo comandos duplicados ou o abandono da interação.

4.5 Análise Temporal e Estabilidade

Para avaliar a consistência operacional a longo prazo, foi conduzida uma análise de séries temporais onde as métricas foram processadas em função da sequência de execução dos testes. O gráfico da Figura 7 foi gerado agrupando-se os resultados pelo identificador sequencial da pergunta (Question ID), simulando o decorrer do tempo e permitindo a comparação direta entre as arquiteturas para uma mesma tarefa.

4.5.1 Latência e Vazão (Gráficos Superiores). Os gráficos superiores focam em métricas instantâneas médias por interação. O gráfico Tempo de Resposta Médio (Superior Esquerda) calcula a média aritmética do tempo total de execução para cada ID. Observa-se que a arquitetura de referência (linha azul) apresenta oscilações severas, atingindo repetidamente o teto de timeout (300s). Em contrapartida, a arquitetura proposta (linha laranja) mantém uma latência basal reduzida, com picos pontuais controlados apenas quando há dependência de latência de rede externa (APIs).

No canto superior direito, o gráfico de Throughput detalha a velocidade de geração textual ($\frac{tokens}{segundo}$). Enquanto o modelo de referência permanece estagnado em baixas taxas (<2 tokens/s), a arquitetura proposta exibe picos de alta vazão, chegando a ultrapassar 25 tokens/s. Isso ocorre porque, ao delegar o raciocínio para agentes especializados com contextos otimizados, o modelo consegue acelerar a fase de síntese da resposta.

4.5.2 Confiabilidade e Eficiência (Gráficos Inferiores). A estabilidade do sistema é analisada no gráfico de Erros Acumulados (Inferior Esquerda), gerado a partir da soma cumulativa de falhas por ID. A abordagem de referência apresenta uma curva de crescimento linear, demonstrando que falhas ocorrem com frequência constante ao longo de todo o teste. Em contraste, a arquitetura proposta atinge a estabilidade rapidamente; a linha permanece plana (próxima de zero), comprovando que o sistema não degrada seu desempenho nem acumula estados de erro, mesmo após extensas baterias de perguntas.

Por fim, o gráfico de Custo Acumulado (Inferior Direita) reflete a eficiência financeira, calculada com base na tabela de preços dos tokens processados. A inclinação acentuada da linha azul demonstra que o modelo monolítico é economicamente ineficiente: devido à redundância de tentativas e ao contexto excessivo, ele consome mais recursos para entregar menos resultados. A arquitetura proposta, representada pela curva suave em laranja, comprova ser economicamente superior, executando as tarefas com o mínimo de tokens necessários e traduzindo eficiência técnica diretamente em economia operacional.

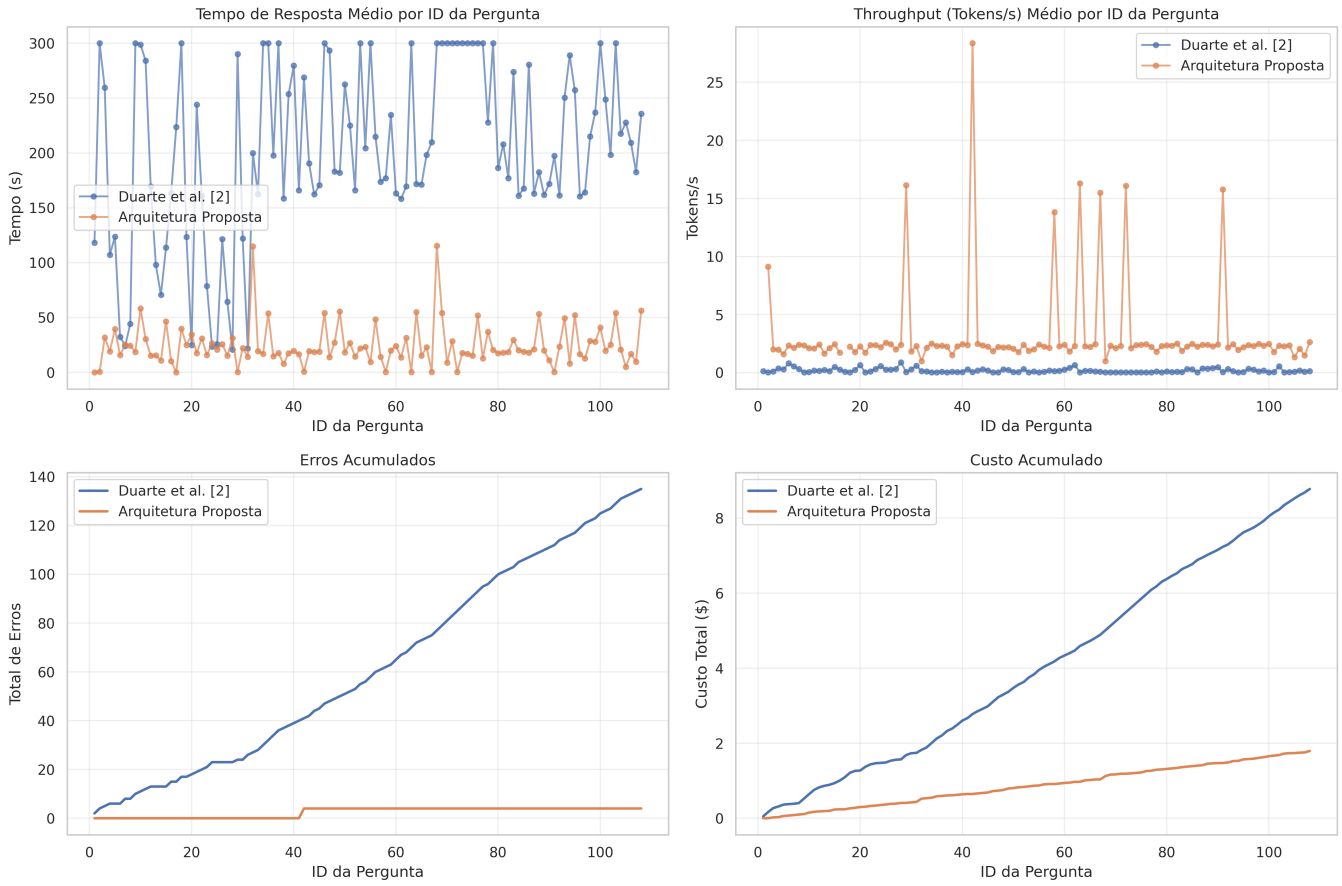

Figure 7: Painel consolidado de desempenho (Grid 2x2). Superior Esquerda: Média do tempo de resposta por ID (limite de timeout em 300s); Superior Direita: Throughput médio (tokens/s), evidenciando picos de agilidade na proposta; Inferior Esquerda: Soma cumulativa de erros, contrastando o crescimento linear da referência com a estabilidade da proposta; Inferior Direita: Soma cumulativa de custo financeiro.

4.6 Correlação de Tempos

A análise de correlação, apresentada na Figura 8, aponta proporcionalidade direta entre a complexidade da entrada e o tempo de processamento em ambos os cenários [21, 27]. O coeficiente de Pearson obtido (r = 0.36) indica forte correlação positiva, demonstrando que tarefas que exigem maior processamento lógico no modelo monolítico, como aquelas envolvendo condicionais múltiplas, resultam em incrementos de latência análogos na arquitetura modular. Os dados sugerem que a complexidade intrínseca da tarefa atua como variável preponderante na determinação do tempo de resposta, independentemente da topologia do sistema adotada.

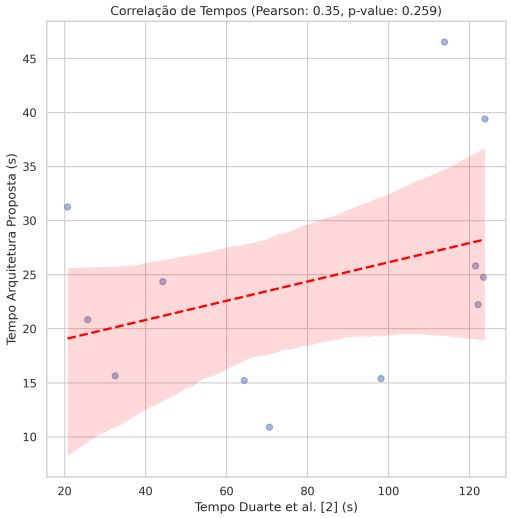

Figure 8: Correlação de latência entre as arquiteturas

Entretanto, a análise dos outliers revela diferença crucial: enquanto a arquitetura monolítica apresenta casos extremos de latência (>300s) em 15% das requisições complexas, a abordagem modular limita esses casos a menos de 1%, devido ao mecanismo de timeout por agente que previne travamentos globais do sistema.

5 Conclusão e Trabalhos Futuros

Este estudo analisou o desempenho de diferentes arquiteturas de interação para assistentes voltados a idosos, contrastando uma abordagem centralizada com uma proposta distribuída. A análise comparativa evidenciou que a arquitetura monolítica, embora funcional e adequada para fluxos puramente conversacionais, tende a apresentar variabilidade de latência quando acumula funções sociais e executivas. Em contrapartida, a abordagem multi-agente baseada em grafos demonstrou-se tecnicamente robusta para operação contínua, estabilizando o desempenho temporal e assegurando elevada taxa de êxito em tarefas sensíveis, mantendo a fluidez necessária para a interação social.

Como continuidade deste trabalho, propõe-se a incorporação de mecanismos de memória de longo prazo, permitindo ao sistema reter preferências, rotinas e padrões de interação do usuário, com controle explícito de atualização. A personalização dinâmica da personalidade do agente também constitui uma direção relevante, possibilitando o ajuste de tom e proatividade conforme o perfil cognitivo do idoso. Outros desdobramentos incluem a adaptação automática dos limiares de validação, a integração de métricas fisiológicas e a avaliação de escalabilidade em cenários multiusuário. Por fim, recomenda-se a validação longitudinal em ambientes reais para mensurar a aceitação e o impacto no uso contínuo do sistema.

6 Materiais Complementares

Para assegurar a reprodutibilidade e a transparência metodológica, disponibilizamos em repositório público todo o código-fonte (arquitetura proposta e replicação de Duarte et al.), o dataset experimental com 108 interações, os logs brutos de execução e os scripts responsáveis pela geração dos gráficos estatísticos. O acervo completo pode ser acessado em: https://github.com/murimuri2/Multi-Agente-e-Roteamento-Determin-stico.git

7 Agradecimentos

Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) Programa de Excelência Acadêmica (PROEX).

References

[1] S. Dos Santos Faria and G. De Araújo Almeida. Envelhecimento populacional e os desafios do cuidado integral ao idoso. Periódicos Cedigma, 1(1):20–26, 2025.

[2] Luize Cunha Duarte, Luan Matheus Trindade Dalmazo, Gabriel Pontarolo, Felipe Gustavo Bombardelli, and Eduardo Todt. Both of us: Development of a multi featured service robot. In 2025 Brazilian Symposium on Robotics (SBR) and 2025 Workshop on Robotics in Education (WRE), pages 285–290, 2025.

[3] Y. Yang, C. Wang, X. Xiang, and R. An. Ai applications to reduce loneliness among older adults: A systematic review of effectiveness and technologies. Healthcare, 13(5):446, 2025.

[4] R. Imran and S.S. Khan. A systematic review on the efficacy of artificial intelligence in geriatric healthcare: a critical analysis of current literature. BMC Geriatrics, 25:248, 2025.

[5] V.S. Lorencini and colaboradores. Perspectivas tecnológicas para o envelhecimento populacional: O benefício da inteligência artificial em idosos. Brazilian Journal of Implantology and Health Sciences, 6, 2024.

[6] J. S. Delgado and G. J. Kölling. Inteligência artificial e machine learning na atenção à saúde da pessoa idosa. Revista de Direito Sanitário, 2025.

[7] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. Advances in Neural Information Processing Systems, 30, 2017.

[8] Hongru Wang, Lingzhi Wang, Yiming Du, Liang Chen, Jingyan Zhou, Yufei Wang, and Kam-Fai Wong. A survey of the evolution of language model-based dialogue systems: Data, task and models. arXiv preprint arXiv:2311.16789, 2023.

[9] S.M. Mohammadabadi, B.C. Kara, C. Eyupoglu, C. Uzay, M.S. Tosun, and O. Karaku. A survey of large language models: Evolution, architectures, adaptation, benchmarking, applications, challenges, and societal implications. Electronics, 14(18):3580, 2025.

[10] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805, 2018.

[11] Xiaoyi Du et al. Enhancing early detection of cognitive decline in the elderly: a comparative study utilizing large language models in clinical notes. eBioMedicine, 109:105401, 2024.

[12] Rumeng Li, Xun Wang, Dan Berlowitz, Jesse Mez, Honghuang Lin, and Hong Yu. Care-ad: a multi-agent large language model framework for alzheimer's disease prediction using longitudinal clinical notes. npj Digital Medicine, 8(1):541, 2025.

[13] Behnam Faraziani and Sevil Eken. Multi-agent ai system for adaptive cognitive training in elderly care. In Proceedings of the 17th International Conference on Agents and Artificial Intelligence (ICAART), volume 1, pages 482–489. SciTePress, 2025.

[14] Fábio Souza, Rodrigo Nogueira, and Roberto Lotufo. Bertimbau: Pretrained bert models for brazilian portuguese. In Intelligent Systems, pages 403–417. Springer International Publishing, 2020.

[15] Caíque B. Fortunato, Ricardo B. C. Costa, and Raquel O. Prates. "isso é de verdade?": Interação de idosos com conteúdos digitais gerados por ia e tecnologias emergentes. In Anais Estendidos do XXIV Simpósio Brasileiro sobre Fatores Humanos em Sistemas Computacionais (IHC). Sociedade Brasileira de Computação, 2025.

[16] Zsolt Nagy. Regex Quick Syntax Reference: Understanding and Using Regular Expressions. Apress, 2018.

[17] David A. Patterson and John L. Hennessy. Computer Organization and Design: The Hardware/Software Interface. Morgan Kaufmann, 5 edition, 2014.

[18] Amey Agrawal, Nitin Kedia, Ashish Panwar, Jayashree Mohan, Nipun Kwatra, Bhargav S. Gulavani, Alexey Tumanov, and Ramachandran Ramjee. Taming throughput-latency tradeoff in LLM inference with Sarathi-Serve. In 18th USENIX Symposium on Operating Systems Design and Implementation (OSDI 24), pages 117–134, Santa Clara, CA, 2024. USENIX Association.

[19] Tom Fawcett. An introduction to roc analysis. Pattern Recognition Letters, 27(8):861–874, 2006.

[20] David M. W. Powers. Evaluation: From precision, recall and f-measure to roc, informedness, markedness and correlation. Journal of Machine Learning Technologies, 2(1):37–63, 2011.

[21] Karl Pearson. Mathematical contributions to the theory of evolution. iii. regression, heredity, and panmixia. Philosophical Transactions of the Royal Society of London. Series A, 187:253–318, 1896.

[22] Douglas C. Montgomery and George C. Runger. Applied Statistics and Probability for Engineers. Wiley, 7 edition, 2018.

[23] George E. P. Box, Gwilym M. Jenkins, Gregory C. Reinsel, and Greta M. Ljung. Time Series Analysis: Forecasting and Control. Wiley, 5 edition, 2015.

[24] Raj Jain. The Art of Computer Systems Performance Analysis: Techniques for Experimental Design, Measurement, Simulation, and Modeling. Wiley, 1991.

[25] Y. C. Tay. Analytical Performance Modeling for Computer Systems. Springer, 2022.

[26] Zixuan Zhou, Xuefei Ning, Ke Hong, Tianyu Fu, Jiaming Xu, Shiyao Li, Yuming Lou, Luning Wang, Zhihang Yuan, Xiuhong Li, Shengen Yan, Guohao Dai, Xiao-Ping Zhang, Yuhan Dong, and Yu Wang. A survey on efficient inference for large language models. arXiv preprint arXiv:2404.14294, 2024.

[27] Jiawei Han, Micheline Kamber, and Jian Pei. Data Mining: Concepts and Techniques. Morgan Kaufmann, 3 edition, 2011.