

Interações de Turno Único e de Múltiplos Turnos em Modelos de IA Generativa de Fronteira (LLMs)

Estado, custo, confiabilidade, segurança e governança

SGAEIA Research Series — Artigo 25

Aridio Silva — Pesquisador Independente, Brasil

ORCID 0009-0008-2411-6995

1º de novembro de 2026 · CC BY 4.0

DOI: 10.5281/zenodo.23867630

DOI: 10.5281/zenodo.23867630

Interações de Turno Único e de Múltiplos Turnos em Modelos de IA Generativa de Fronteira (LLMs)

Estado, custo, confiabilidade, segurança e governança

SGAEIA Research Series — Artigo 25
Aridio Silva — Pesquisador Independente, Brasil
Criador do SGAEIA — Secure Governed Autonomous Edge Intelligence Architecture
ORCID: 0009-0008-2411-6995
1º de novembro de 2026
© 2026 Aridio Silva. Texto e ilustrações originais licenciados sob CC BY 4.0.

Resumo

Este artigo examina como a passagem de uma interação de turno único para uma trajetória multi-turno altera o estado, o custo, a confiabilidade, a segurança e a governança de sistemas baseados em modelos de linguagem de grande porte. A análise é uma revisão narrativa crítica apoiada em artigos científicos, documentação oficial de provedores, normas, taxonomias de segurança e cálculos próprios explicitamente identificados. O modelo não possui memória conversacional intrínseca entre inferências; a continuidade é reconstruída pelo cliente ou por serviços de estado do provedor. Quando a aplicação reapresenta integralmente um histórico que cresce aproximadamente a uma taxa constante, a soma lógica dos tokens de entrada cresce quadraticamente com o número de chamadas, embora cache, truncamento, compactação e memória externa alterem custo faturado, latência e computação efetiva.

Evidências de benchmarks mostram que fragmentação de requisitos, dependências entre turnos, mudanças de intenção, recuperação de informação e dependência de respostas anteriores produzem falhas distintas; não existe uma única condição "multi-turno". A evidência sobre raciocínio explícito e técnicas de memória é mista e depende da tarefa e do protocolo. Em sistemas agênticos, o histórico também amplia ataques graduais, persistência de conteúdo malicioso, propagação de erro e consumo descontrolado. O artigo relaciona esses resultados ao roteamento governado e ao SGAEIA, defendendo a separação entre transcrição, estado operacional e evidência de assurance. Essa relação constitui interpretação arquitetural e agenda de validação, não prova empírica do framework.

Palavras-chave: LLM; turno único; multi-turno; estado conversacional; roteamento de LLM; cache de prompt; confiabilidade; segurança; governança em tempo de execução; SGAEIA.

Sumário

1. Introdução, escopo e método
2. Fundamentos: turno, estado e mecânica
3. Histórico e evolução dos métodos e técnicas
4. Desempenho em multi-turno: a evidência
5. Economia de tokens e créditos
6. Segurança e risco em multi-turno
7. Estado da arte em gerenciamento de estado e mitigação de deriva
8. Arquitetura de conformidade e o framework SGAEIA
9. Confiabilidade da informação, verificação e auditoria
10. Estado da arte e perspectivas futuras
11. Conclusão

1. Introdução, escopo e método

Definições. Mensagem é uma unidade atribuída a um papel, como usuário, assistente, sistema ou ferramenta. **Chamada de inferência** é uma execução do modelo que recebe uma representação do estado e produz uma saída. **Rodada de interação** é, neste artigo, o ciclo formado por uma nova entrada do usuário e a resposta correspondente do sistema, podendo incluir chamadas internas de ferramentas. Nas equações de custo, T representa o número de chamadas principais ou rodadas, conforme indicado; quando um benchmark adotar outra convenção, ela será declarada explicitamente.

Pergunta de pesquisa. Como a escolha entre turno único e multi-turno afeta (a) a mecânica de estado, (b) o custo em tokens e créditos, (c) o desempenho e a confiabilidade, e (d) a segurança e a governança, nos LLMs de fronteira?

Método e política de fontes. Priorizam-se fontes primárias: artigos (arXiv, anais), documentação e blogs oficiais dos provedores, normas e legislação. Fontes secundárias só entram quando sinalizadas. Cada referência foi conferida quanto a autoria, data, identificador e venue; o resultado da auditoria está na Seção 12 (Bibliografia), com os itens **não verificados** marcados como tais. Preços e níveis de raciocínio são um *retrato* de out./nov. de 2026, retirado das páginas oficiais, e mudam com frequência.

Limitações. Nenhum provedor publica o desempenho de cada modelo e de cada nível de raciocínio separado entre turno único e multi-turno. As tabelas de custo usam premissas explícitas e são **cálculo próprio**, não dado dos provedores.

Como ler. Cada seção termina com um quadro de análise. As equações estão em imagens numeradas; as figuras trazem descrição textual.

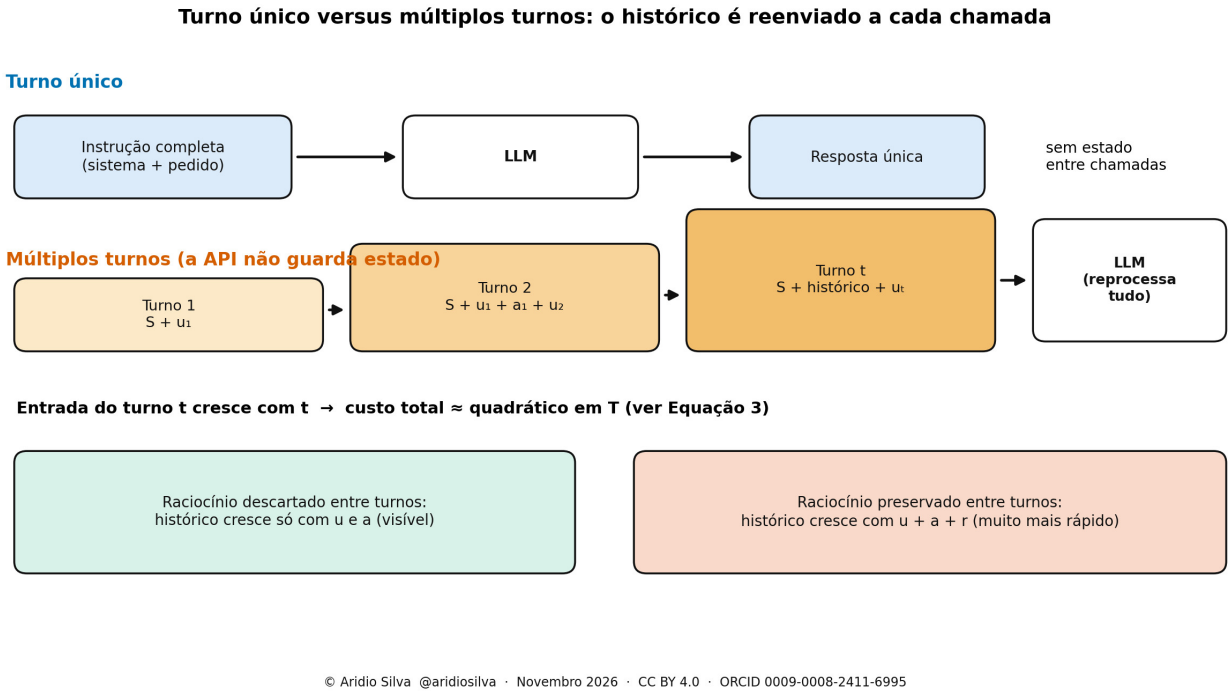
2. Fundamentos: turno, estado e mecânica

2.1 O modelo não possui memória conversacional intrínseca

Um modelo de linguagem não preserva, por si mesmo, memória conversacional entre inferências independentes. A continuidade pode ser reconstruída pelo cliente, que reapresenta mensagens ou um estado derivado, ou por um serviço do provedor, que mantém itens, identificadores ou mecanismos de compactação no servidor. Em ambos os casos, a resposta atual depende de uma representação selecionada do passado. Essa representação deve ser tratada como entrada manipulável, sujeita a truncamento, resumo, erro, controle de acesso e auditoria, e não como registro automaticamente confiável. Três consequências práticas:

- **Custo e latência** crescem a cada turno, porque o histórico é reprocessado, e a conversa fica limitada pela janela de contexto.
- **O cliente controla o "passado".** Turnos anteriores podem ser sintéticos; para segurança, o histórico é uma superfície de entrada manipulável e não um registro confiável.

- **Multi-turno não é o mesmo que contexto longo.** Um prompt único de 100 mil tokens é contexto longo em turno único; o que define o multi-turno é a revelação incremental de informação e a dependência das próprias respostas anteriores do modelo.



© Aridio Silva @aridiosilva · Novembro 2026 · CC BY 4.0 · ORCID 0009-0008-2411-6995

Figura 1. Esquema de turno único (uma chamada, sem estado) e de múltiplos turnos (histórico reenviado e crescente). Elaboração do autor.

2.2 Turno único: eficiência isolada

No modelo de turno único a interação restringe-se a uma requisição e uma resposta, e o modelo não possui memória inerente de interações anteriores.

- **Isolamento e determinismo relativo.** Cada chamada é um evento atômico, o que facilita teste unitário de prompts, reprodutibilidade e políticas estáticas de validação de entrada e saída.
- **Baixa latência e custo.** Sem acumulação de histórico, o consumo de tokens de entrada é mínimo (o cache de prompt é tratado à parte), o que reduz custo por inferência e o *Time to First Token*.
- **Menor persistência entre chamadas.** A ausência de histórico reduz a persistência de conteúdo malicioso entre inferências, mas não elimina injeção indireta em uma única chamada que contenha páginas, e-mails, documentos ou resultados RAG não confiáveis.

Limitações arquiteturais: ausência de refinamento iterativo (qualquer ajuste exige reescrever o prompt) e inviabilidade para tarefas que pedem planejamento passo a passo, depuração iterativa ou exploração do espaço de design.

2.3 Multi-turno: estado e gestão de contexto

No multi-turno o histórico acumulado (usuário, assistente e, opcionalmente, chamadas de ferramentas) é reprocessado ou gerido a cada nova interação. Em sessões longas, três estratégias evitam saturar a janela e degradar a atenção:

- **Janela deslizante (*sliding window*).** Mantém apenas as N últimas interações; simples, mas sofre de "amnésia catastrófica" para dependências de longo prazo.
- **Sumarização dinâmica.** Um subsistema (ou o próprio LLM em segundo plano) comprime o histórico antigo em resumos consolidados.
- **Memória externa / RAG conversacional.** Histórico e artefatos ficam em bancos vetoriais ou relacionais e são recuperados por busca semântica conforme a intenção do turno atual.

Uma linha de pesquisa relacionada trata a janela de contexto como memória principal de um sistema operacional. O MemGPT [6] propõe *gestão de contexto virtual*: o contexto principal (instruções, *working context* e fila FIFO) é complementado por armazenamento externo, com paginação comandada por chamadas de função.

2.4 Complexidade: o que é quadrático e o que não é

Nos Transformers [1], a atenção tem custo $O(n^2 \cdot d)$ por camada (Equação 1). O crescimento quadrático é polinomial, não exponencial. Há ainda duas camadas a distinguir:

1. **Dentro de uma chamada**, o pré-preenchimento (*prefill*) do contexto tem custo quadrático no tamanho da entrada, e cada token gerado atende às chaves já em cache, com custo linear no contexto.
2. **Ao longo da conversa**, se cada rodada acrescenta quantidade aproximadamente constante de tokens e a aplicação reapresenta integralmente o histórico, sem truncamento ou compactação, a soma lógica dos tokens de entrada cresce como $O(T^2)$ no número de chamadas T (Equações 2 e 3). Cache, persistência no servidor e políticas de cobrança podem fazer o custo faturado e a computação efetiva seguirem curvas diferentes.

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^\top}{\sqrt{d_k}}\right)V \qquad \text{custo por camada} = O(n^2\, d)$$

© 2026 Aridio Silva · @aridiosilva · Novembro de 2026 · CC BY 4.0 · ORCID 0009-0008-2411-6995

Equação 1. Atenção escalonada por produto escalar e complexidade por camada, conforme Vaswani et al. [1].

$$I_t = S + tu + (t-1)(a + \kappa r)$$

© 2026 Aridio Silva · @aridiosilva · Novembro de 2026 · CC BY 4.0 · ORCID 0009-0008-2411-6995

Equação 2. Entrada do turno t, em que S é o prompt de sistema, u os tokens do usuário por turno, a os tokens visíveis do assistente, r os tokens de raciocínio e κ = 1 se o raciocínio é mantido no histórico (κ = 0 caso contrário).

$$\sum_{t=1}^{T} I_t = TS + u\frac{T(T+1)}{2} + (a + \kappa r)\frac{T(T-1)}{2} = O(T^2)$$

© 2026 Aridio Silva · @aridiosilva · Novembro de 2026 · CC BY 4.0 · ORCID 0009-0008-2411-6995

Equação 3. Entrada acumulada em T turnos: quadrática em T.

2.5 Matriz comparativa

Tabela 1

Critério	Turno único (single-turn)	Multi-turno (multi-turn)
Complexidade de estado	Sem continuidade intrínseca; estado externo opcional	Continuidade reconstruída pelo cliente ou por serviço do provedor
Custo de inferência	Baixo e linear por requisição	Crescente (reprocessamento do histórico; ~T^2)
Latência	Mínima	Maior (escala com o tamanho do contexto)
Superfície de risco	Isolada e menor	Acumulativa (poluição de contexto e de estado)
Casos ideais	Classificação, extração, tradução, pipelines estáticos	Assistência interativa, geração de código, agentes autônomos
Revelação de informação	Completa no início	Incremental; permite subespecificação

© 2026 Aridio Silva · SGAEIA Research Series — Article 25 · CC BY 4.0

Tabela 1 do Artigo 25. © 2026 Aridio Silva · SGAEIA Research Series — Article 25 · CC BY 4.0

2.6 Análise da seção

Tabela 2

Dimensão	Análise
Implicações	O estado não vive intrinsecamente no modelo; pode ser mantido pelo cliente ou por serviço do provedor. Quem projeta o sistema decide o que reenviar, resumir, descartar e auditar.
Pontos fortes	Turno único: simplicidade, custo previsível, testabilidade. Multi-turno: refinamento iterativo, exploração de requisitos incompletos, agentes.
Pontos fracos	Turno único: não cobre tarefas exploratórias. Multi-turno: custo ~T^2, latência crescente, risco acumulativo e dependência de gestão de histórico.
Trade-offs	Capacidade de refinar versus custo e risco. Janela deslizante é barata e perde dependências; sumarização preserva mais e pode distorcer; memória externa escala e adiciona complexidade e superfície de ataque.

© 2026 Aridio Silva · SGAEIA Research Series — Article 25 · CC BY 4.0

Tabela 2 do Artigo 25. © 2026 Aridio Silva · SGAEIA Research Series — Article 25 · CC BY 4.0

3. Histórico e evolução dos métodos e técnicas

A evolução recente de LLMs pode ser lida em **sete dimensões** que se influenciam mutuamente. A Figura 2 apresenta somente marcos com data conferida em fonte primária (Seção 12); marcos sem verificação foram deixados de fora de propósito.

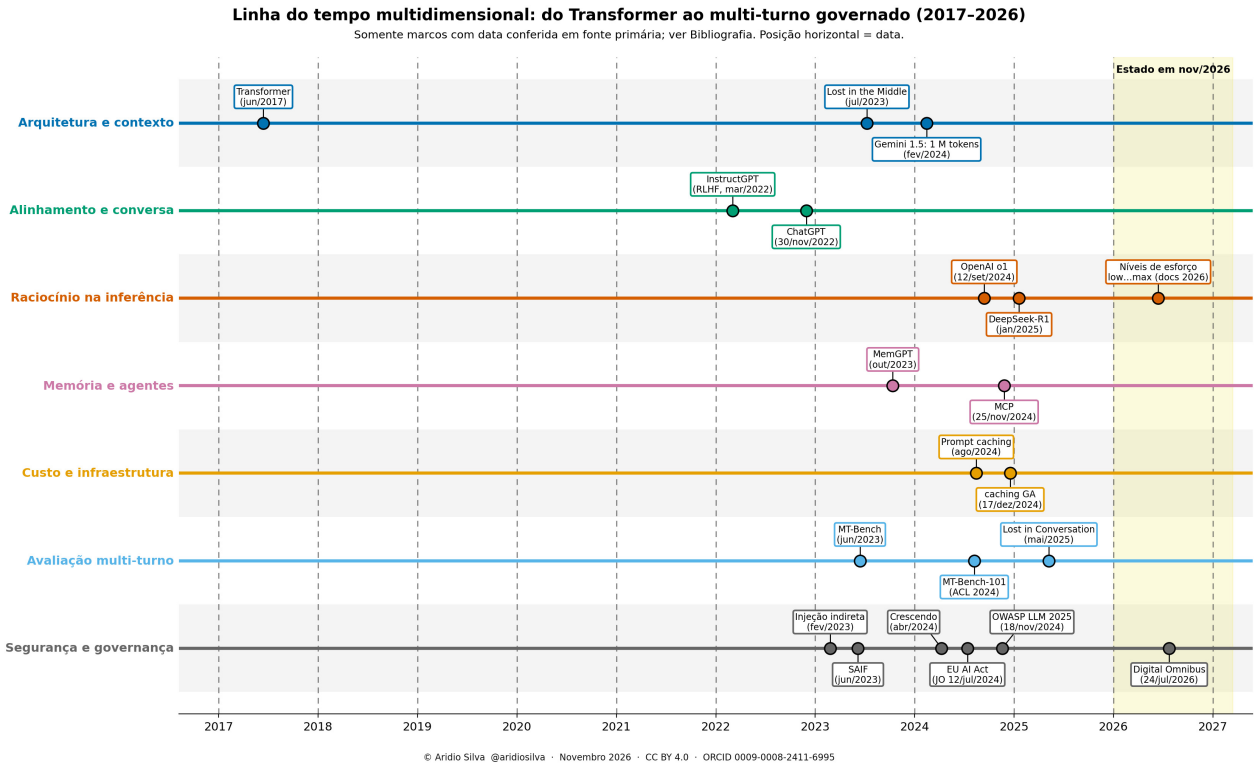


Figura 2. Linha do tempo multidimensional (arquitetura e contexto; alinhamento e conversa; raciocínio na inferência; memória e agentes; custo e infraestrutura; avaliação multi-turno; segurança e governança). Elaboração do autor a partir das fontes [1]–[22].

3.1 Era 1 (2017–2021): arquitetura e prompting

O Transformer [1], publicado em junho de 2017, substituiu recorrência por atenção e tornou o treinamento paralelizável. Os modelos eram treinados com janelas pequenas, entre 512 e 2.048 tokens [5], e usados por *prompting*: dava-se um texto ou alguns exemplos e o modelo completava [2]. Esse é, em essência, um regime de turno único.

3.2 Era 2 (2022): alinhamento e o formato de diálogo

O InstructGPT [2] (mar./2022) mostrou que ajustar o modelo com feedback humano (RLHF) o torna preferível a um GPT-3 mais de cem vezes maior, em instruções do usuário. Em 30 de novembro de 2022 a OpenAI lançou o ChatGPT, cujo formato de diálogo permite responder perguntas de acompanhamento, admitir erros e contestar premissas incorretas [18]. Aqui o multi-turno deixa de ser tema de pesquisa e se torna **formato de produto**.

3.3 Era 3 (2023): contexto, memória, avaliação e primeiros ataques

- **Contexto.** Em 2023 já havia modelos com 4 mil a 100 mil tokens, mas *Lost in the Middle* [5] (jul./2023) mostrou que o desempenho cai quando a informação relevante está no meio e que mais contexto nem sempre melhora o resultado.
- **Memória.** O MemGPT [6] (out./2023) propôs paginação de contexto e armazenamento externo.
- **Avaliação.** O MT-Bench [4] (jun./2023) instituiu a avaliação de diálogo em dois turnos com juiz-LLM.
- **Segurança.** Greshake et al. [3] (fev./2023) introduziram a *injeção indireta de prompt*: dados recuperados passam a agir como "código arbitrário". O Google lançou o SAIF [17] (jun./2023).

3.4 Era 4 (2024): contexto longo, cache, raciocínio e protocolos

- **Contexto longo.** O Gemini 1.5 [22] (15/fev./2024) ofereceu janela experimental de 1 milhão de tokens (padrão de 128 mil); o próprio Google reconhece custo computacional elevado e necessidade de otimizar latência.
- **Cache.** O *prompt caching* da Anthropic [20] (ago./2024; disponibilidade geral em dez./2024) reportou até 90% menos custo e 85% menos latência em prompts longos; em conversa de 10 turnos com prompt de sistema longo, a latência caiu de ~10 s para ~2,5 s e o custo, 53%.
- **Raciocínio.** O o1 [19] (12/set./2024) treinou o modelo por reforço a produzir uma cadeia de pensamento antes de responder, com desempenho crescente em função do tempo de pensamento na inferência (*test-time compute*).
- **Agentes e protocolos.** O MCP [21] (25/nov./2024) padronizou a conexão entre assistentes e ferramentas/dados.
- **Avaliação e segurança.** O MT-Bench-101 [7] (ACL 2024) criou taxonomia hierárquica de 13 tarefas; o *Crescendo* [8] mostrou jailbreak multi-turno com entradas benignas; o OWASP publicou a edição 2025 do Top 10 para LLMs [16]; o Regulamento (UE) 2024/1689 [14] foi publicado em 12/jul./2024.

3.5 Era 5 (2025–2026): confiabilidade, esforço de raciocínio e recalibração regulatória

- O DeepSeek-R1 [11] (jan./2025) difundiu o raciocínio treinado por reforço em modelos abertos.
- Laban et al. [10], publicado na ICLR 2026 e distinguido como *Outstanding Paper*, mostrou degradação em todos os modelos avaliados no protocolo de conversas subespecificadas; os dois modelos de raciocínio testados não eliminaram a perda.
- Em 2026, os provedores expõem **níveis de esforço** de raciocínio (low a max) como parâmetro de custo e qualidade [24]–[29] (Seção 5).
- O Regulamento (UE) 2026/1744 [15], em vigor desde 27/jul./2026, adiou as obrigações de alto risco do Anexo III para 2/dez./2027.

3.6 Leitura transversal da linha do tempo

Três padrões se destacam. (1) Cada avanço de **capacidade** (contexto longo, raciocínio) foi seguido por avanços de **custo** (cache) e de **avaliação**, mas a avaliação multi-turno chegou depois do produto. (2) A **segurança** acompanhou a conversa: o risco migrou do prompt isolado para o histórico e para os dados recuperados. (3) A **governança** passou de princípios para controles em tempo de execução, tema das Seções 7 e 8.

3.7 Análise da seção

Tabela 3

Dimensão	Análise
Implicações	O multi-turno tornou-se um formato amplamente adotado em produtos conversacionais, mas foi avaliado, precificado e protegido depois de adotado. Isso explica a lacuna entre o desempenho de laboratório (turno único) e o de uso real.
Pontos fortes	Ritmo rápido de inovação; mitigação de custo (cache) e de memória (paginação) surgiram em poucos anos.
Pontos fracos	Muitos marcos de 2026 não têm data pública verificável por fonte primária; a linha do tempo é deliberadamente conservadora. Benchmarks envelhecem rápido.
Trade-offs	Mais contexto e mais raciocínio elevam capacidade, custo e superfície de ataque ao mesmo tempo.

© 2026 Aridio Silva · SGAEIA Research Series — Article 25 · CC BY 4.0

Tabela 3 do Artigo 25. © 2026 Aridio Silva · SGAEIA Research Series — Article 25 · CC BY 4.0

4. Desempenho em multi-turno: a evidência

4.1 Benchmarks que introduziram o multi-turno

O MT-Bench [4] avaliou conversas de dois turnos (por exemplo, pedir um texto e depois reexplicá-lo para uma criança); o MT-Bench-101 [7] ampliou para 4.208 turnos em 1.388 diálogos e 13 tarefas, organizados em percepção, adaptabilidade e interatividade, e observou que técnicas usuais de alinhamento e desenhos específicos para chat não deram ganhos evidentes nas capacidades multi-turno. Esses benchmarks são, em geral, **episódicos**: cada turno traz uma subtarefa avaliável isoladamente.

4.2 Lost in Conversation (Laban et al., ICLR 2026)

Laban et al. [10] (Microsoft Research e Salesforce Research) comparam turno único e multi-turno **sobre as mesmas tarefas**. O método de *sharding* fragmenta cada instrução em partes, e o usuário simulado revela no máximo uma por turno. Três condições principais: **FULL** (instrução completa em um turno), **CONCAT** (fragmentos concatenados em um turno) e **SHARDED** (um fragmento por turno).

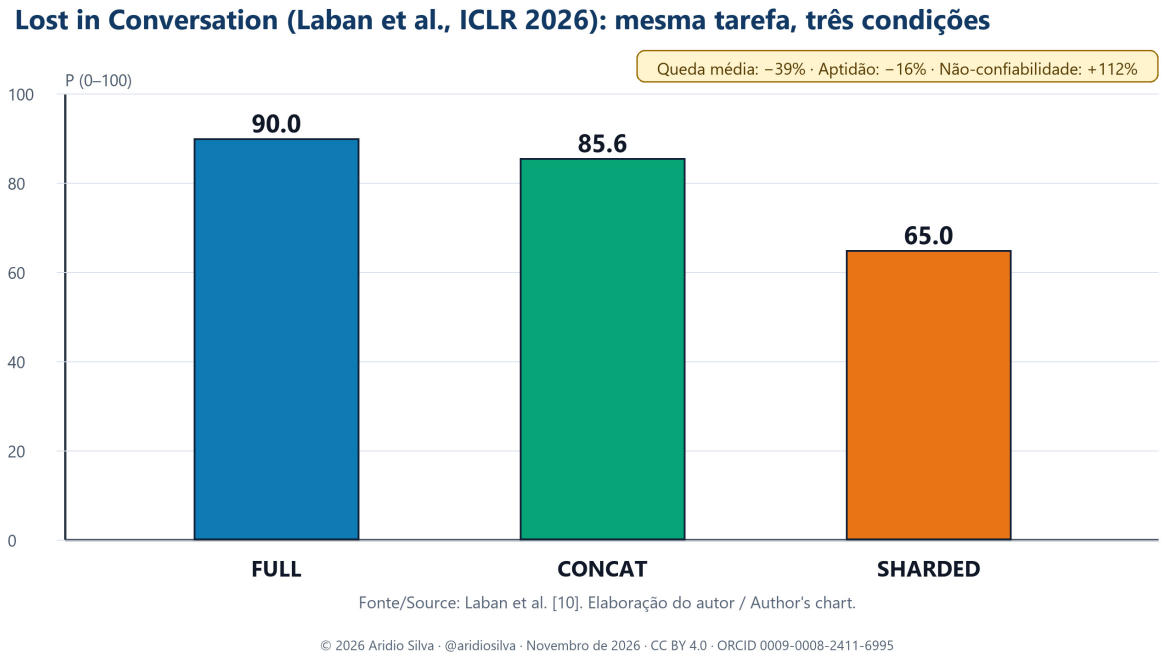


Figura 3. Desempenho médio P (0 a 100) nas três condições, segundo [10]. Barras: 90 (FULL), 85,6 (CONCAT = 95,1% de FULL) e 65 (SHARDED). Elaboração do autor.

Métricas. Para cada instrução, com N simulações e escores S:

$$P = (1/N)\ \Sigma_i\ S_i$$
$$A^{90} = \text{percentile}_{90}(S)$$
$$U^{90} = \text{percentile}_{90}(S) - \text{percentile}_{10}(S)$$

S = escores em N simulações

© 2026 Aridio Silva · @aridiosilva · Novembro de 2026 · CC BY 4.0 · ORCID 0009-0008-2411-6995

Equação 4. Desempenho médio (P), aptidão (A^{90}, o desempenho de melhor caso) e não-confiabilidade (U, a distância entre os percentis 90 e 10) [10].

Resultados (v1, 9/mai./2025). 15 LLMs de oito famílias, seis tarefas (código, banco de dados, ações/APIs, matemática, dados-para-texto e resumo), mais de 200 mil conversas simuladas. Média de 90% em turno único contra 65% em multi-turno; queda de 39% no resumo e no texto; aptidão −16% e não-confiabilidade +112%. A degradação aparece já com **dois** turnos. A condição CONCAT atinge 95,1% de FULL, logo a perda **não** vem da reformulação. Os dois modelos de raciocínio testados (o3 e DeepSeek-R1) pioram de modo semelhante e geram respostas em média 33% mais longas, com mais suposições.

Causas qualitativas: tentativas prematuras de resposta final; "inchaço" das respostas (respostas finais 20 a 300% maiores); *perda nos turnos do meio*; e verbosidade. **Temperatura zero não resolve:** com temperatura 0 de ambos os lados ainda restam cerca de 30 pontos de não-confiabilidade. **Mitigações:** repetir os requisitos a cada turno (SNOWBALL) recupera 15 a 20% da queda; o RECAP final é melhor, mas irrealista.

Limitações do estudo (apontadas pelos autores): simulação automática, tarefas analíticas, apenas inglês e texto. Os autores consideram que a degradação real tende a ser **maior** que a medida. **Recomendações:** tentar de novo em uma conversa nova e **consolidar** os requisitos num único prompt.

4.3 Contexto longo e atenção: Lost in the Middle

Liu et al. [5] mostraram curva em U: o desempenho é maior quando a informação relevante está no início ou no fim do contexto e cai quando está no meio, e diminui conforme o contexto cresce, mesmo em modelos de contexto longo. Laban et al. observaram um análogo em turnos (*loss-in-middle-turns*).

4.4 Evidência convergente e divergente em benchmarks multi-turno

LLMs Get Lost in Multi-Turn Conversation não é evidência isolada. A versão final foi publicada na ICLR 2026 e recebeu a distinção *Outstanding Paper* [37], [38]. Seu resultado principal — queda média de 39% quando os requisitos da mesma tarefa são revelados gradualmente — deve ser interpretado dentro do desenho de *sharded simulation*: tarefas analíticas, conversas simuladas, texto em inglês e modelos de uma janela temporal específica.

Outros benchmarks medem falhas parcialmente diferentes. **MT-Eval** organiza interações em recordação, expansão, refinamento e acompanhamento [39]. **MultiChallenge**, publicado em *Findings of ACL 2025*, avalia quatro desafios realistas que exigem simultaneamente seguimento de instruções, alocação de contexto e raciocínio em contexto; todos os modelos avaliados ficaram abaixo de 50% de acurácia, com melhor média de 41,4% [40]. **StructFlowBench** modela seis relações estruturais entre turnos e identifica dificuldades na compreensão do fluxo da conversa [41]. **mtRAG**, publicado em TACL, acrescenta recuperação de informação por meio de 110 conversas humanas, média de 7,7 turnos e 842 tarefas [42].

A evidência sobre raciocínio explícito é mista. No desenho de Laban et al., os dois modelos de raciocínio avaliados não eliminaram a degradação e produziram respostas mais longas. Em contraste, **MARS-Bench** relata que raciocínio explícito aumenta a robustez em diálogos longos com mudança de intenção e dependências cruzadas [43]. A conclusão defensável não é que "mais raciocínio não funciona", mas que seu efeito depende da tarefa, do protocolo, do gerenciamento de memória e da métrica.

Em conjunto, esses estudos mostram que "multi-turno" não é uma única condição. Pelo menos cinco mecanismos devem ser separados: (1) requisitos revelados em fragmentos; (2) dependências estruturais entre turnos; (3) distração e mudança de intenção; (4) recuperação de evidência externa; e (5) dependência das próprias respostas anteriores do modelo. Uma mitigação eficaz para um deles — como recapitulação, memória compacta ou raciocínio explícito — pode falhar ou aumentar custo em outro.

4.5 Implicações para estado, custo e SGAEIA

Os resultados reforçam a necessidade de representar o estado da tarefa separadamente da transcrição. Requisitos confirmados, fatos validados, decisões revogadas, permissões e resultados de ferramentas não devem depender apenas de sua posição no histórico textual. Uma arquitetura de governança pode manter contratos tipados, proveniência e políticas determinísticas enquanto usa a conversa como interface. Essa relação é uma **interpretação do autor e hipótese arquitetural do SGAEIA**, não uma validação empírica do framework. Sua eficácia deve ser testada contra baselines como replay integral, janela deslizante, resumo rolante e memória recuperada.

4.6 Um modelo ilustrativo de propagação de erro

Para fixar a intuição (e **não** como resultado empírico), suponha que a cada turno exista probabilidade ε de o modelo introduzir uma premissa errada que o usuário aceita (cascata de alucinações). A probabilidade de uma conversa de T turnos sem erro acumulado é dada pela Equação 6. Com ε = 5% e T = 20, resta cerca de 36%.

$$\Pr(\text{sem erro em } T \text{ turnos}) = (1-\varepsilon)^T \approx e^{-\varepsilon T}$$

© 2026 Aridio Silva · @aridiosilva · Novembro de 2026 · CC BY 4.0 · ORCID 0009-0008-2411-6995

Equação 6. Modelo didático de erro composto; ε e T são parâmetros hipotéticos.

4.7 Análise da seção

Tabela 4

Dimensão	Análise
Implicações	O desempenho de turno único superestima o de uso real. A métrica que mais se degrada é a confiabilidade, e não a aptidão.
Pontos fortes	Desenho experimental controlado (mesmas tarefas, três condições, escala de 200 mil conversas) e métricas bem definidas (P, A, U).
Pontos fracos	Modelos de 2025; simulação idealizada; tarefas analíticas em inglês. Não há dado público equivalente para as gerações de 2026 nem para cada nível de esforço.
Trade-offs	Consolidar o pedido num turno aumenta confiabilidade, mas perde a vantagem exploratória do multi-turno; resumos e recapitulações recuperam parte da queda ao custo de mais tokens.

© 2026 Aridio Silva · SGAEIA Research Series — Article 25 · CC BY 4.0

Tabela 4 do Artigo 25. © 2026 Aridio Silva · SGAEIA Research Series — Article 25 · CC BY 4.0

5. Economia de tokens e créditos

5.1 Mecânica de cobrança

A cobrança tem três componentes: tokens de **entrada**, tokens de **saída** (incluindo os de raciocínio, cobrados como saída) e a **leitura de cache** da entrada repetida. Em multi-turno, quase todo o histórico reenviado é lido do cache, por isso o preço de **leitura de cache** diferencia os provedores mais do que o preço de entrada.

$$C = \frac{1}{10^6} \sum_{t=1}^{T} \left[p_w N_t^{\text{novos}} + p_c \left(I_t - N_t^{\text{novos}} \right) + p_o \left(a + r \right) \right]$$

© 2026 Aridio Silva · @aridiosilva · Novembro de 2026 · CC BY 4.0 · ORCID 0009-0008-2411-6995

Equação 5. Custo de uma conversa de T turnos com cache: p_w (escrita), p_c (leitura de cache), p_o (saída), em US$ por milhão de tokens; N_novos é a parte nova da entrada em cada turno.

5.2 Níveis de raciocínio por provedor (retrato de nov./2026)

Tabela 5

Provedor	Modelo	Níveis de raciocínio	Padrão
OpenAI	GPT-6 Astra	low → max	não confirmado
OpenAI	GPT-6.1 Sol	low → max	medium
OpenAI	GPT-6.1 Terra, GPT-6 Luna	none → max	medium (Luna)
Anthropic	Fable 5.1, Sonnet 5.5	low → max	high
Anthropic	Opus 5.5, Haiku 5.5	low → max	medium
Google	Gemini 4 Argon, 3.8 Flash	low, medium, high	medium
Google	3.5 Flash-Lite	minimal → high	minimal
xAI	Grok 4.7	low → xhigh	high
DeepSeek	V4.1-Flash, V4-Pro	com ou sem raciocínio	com raciocínio

© 2026 Aridio Silva · SGAEIA Research Series — Article 25 · CC BY 4.0

Tabela 5 do Artigo 25. © 2026 Aridio Silva · SGAEIA Research Series — Article 25 · CC BY 4.0

A disponibilidade de cada nível varia por modelo; a Anthropic indica a página de effort* como referência [24]. Fontes: [24]–[29]. Os nomes e níveis mudam com frequência; confirme nas páginas oficiais.

5.3 Como cada provedor trata o raciocínio dos turnos anteriores

- **OpenAI** [25]: nos modelos anteriores ao GPT-5.6, o raciocínio dos turnos passados não entra no contexto seguinte; a família GPT-5.6 passou a incluí-lo por padrão, e o parâmetro `reasoning.context` escolhe o comportamento (`all_turns` é suportado pelo GPT-5.6 e pelo GPT-6.1 Sol). Um item `configuration_update` altera o esforço entre respostas sem mudar o prefixo, preservando o cache.
- **Anthropic** [24]: blocos de raciocínio de turnos anteriores que permanecem no contexto são cobrados como entrada (todos os turnos em alguns modelos, só o último nos demais). Mudar o *effort* no topo da requisição **invalida o cache**; a mudança por mensagem (beta) preserva os pontos de cache. O raciocínio por requisição tende a diminuir à medida que a conversa fica longa.
- **Google** [27]: as *thought signatures* mantêm a continuidade do raciocínio; no modo sem estado, os blocos de pensamento devem ser reenviados exatamente como recebidos.
- **xAI** [28]: o grok-4.7 sempre devolve raciocínio criptografado, que deve ser reenviado sem alteração para manter o raciocínio entre turnos; o raciocínio não pode ser desligado.
- **DeepSeek** [29]: modo com e sem raciocínio; preços de pico e fora de pico.

5.4 Preços (US$ por milhão de tokens) e peso do cache

Tabela 6

Modelo	Entrada	Saída	Leitura de cache	Cache / entrada
GPT-6 Astra	10,00	50,00	1,00	10%
GPT-6.1 Sol	2,00	10,00	0,10	5%
GPT-6.1 Terra	1,00	5,00	0,05	5%
GPT-6 Luna	0,10	0,50	0,01	10%
Claude Fable 5.1	10,00	50,00	0,25	2,5%
Claude Opus 5.5	4,00	20,00	0,20	5%
Claude Sonnet 5.5	2,00	10,00	0,10	5%
Claude Haiku 5.5 (≤100 mil)	0,10	0,50	0,01	10%
Gemini 4 Argon (até 30/nov.)	2,00	10,00	0,10	5%
Gemini 3.8 Flash (até 31/dez.)	0,75	3,75	0,075	10%
Grok 4.7 (<200 mil)	2,00	6,00	0,50	25%
DeepSeek V4-Pro (pico)	1,32	3,96	0,044	3,3%
DeepSeek V4.1-Flash (pico)	0,30	1,20	0,006	2%

© 2026 Aridio Silva · SGAEIA Research Series — Article 25 · CC BY 4.0

Tabela 6 do Artigo 25. © 2026 Aridio Silva · SGAEIA Research Series — Article 25 · CC BY 4.0

Ressalvas dos provedores: o Haiku 5.5 tem preço maior acima de 100 mil tokens; o Gemini 4 Argon dobra o preço a partir de 1º/dez./2026 e só oferece cache implícito; o grok-4.7 dobra acima de 200 mil tokens; a DeepSeek cobra metade fora do pico; e os modelos Claude a partir da 4.7 usam um tokenizador que gera cerca de 30% mais tokens para o mesmo texto, o que torna o preço por token **não comparável** 1:1 entre provedores [23]. Fontes: [23], [25], [26], [28], [29].

5.5 Exemplo de 20 turnos (cálculo próprio)

Premissas: S = 2.000, u = 300, a = 700 tokens; raciocínio r = 1.000 (esforço baixo) ou 8.000 (alto); o cache acerta sempre; para este cenário simplificado, a escrita de cache não recebe sobretaxa. Essa premissa não é universal: provedores podem cobrar escrita, TTL ou modalidades distintas de cache. Pela Equação 3, a entrada acumulada em 20 turnos é de **236 mil tokens** (21,3 mil novos e 214,7 mil lidos do cache) com raciocínio descartado; se o raciocínio de 8.000 tokens fosse **mantido** no histórico, a entrada subiria para **1,756 milhão** (7,4 vezes), com 173,3 mil tokens de entrada no 20º turno.

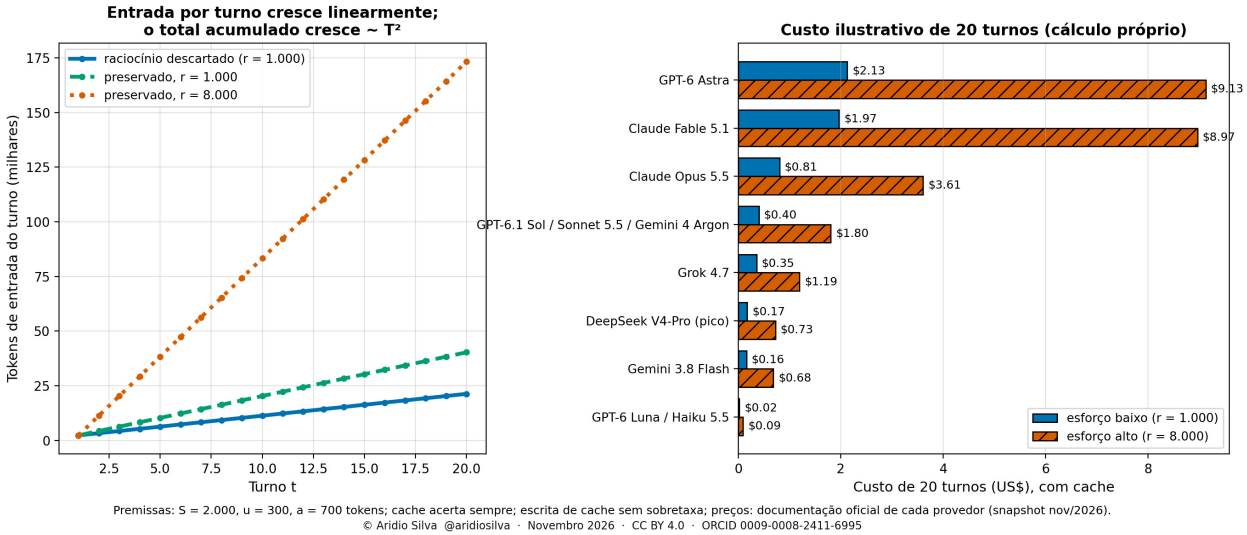


Figura 4. À esquerda, tokens de entrada por turno em três cenários; à direita, custo ilustrativo de 20 turnos por modelo, com cache. Cálculo do autor sobre preços oficiais.

Tabela 7

Modelo	20 turnos, esforço baixo	20 turnos, esforço alto	Sem cache, esforço baixo
GPT-6 Astra	US$ 2,13	US$ 9,13	US$ 4,06
Claude Fable 5.1	US$ 1,97	US$ 8,97	US$ 4,06
Claude Opus 5.5	US$ 0,81	US$ 3,61	US$ 1,62
GPT-6.1 Sol / Sonnet 5.5 / Gemini 4 Argon	US$ 0,40	US$ 1,80	US$ 0,81
Grok 4.7	US$ 0,35	US$ 1,19	US$ 0,68
DeepSeek V4-Pro (pico)	US$ 0,17	US$ 0,73	US$ 0,45
Gemini 3.8 Flash	US$ 0,16	US$ 0,68	US$ 0,30
GPT-6 Luna / Haiku 5.5	US$ 0,02	US$ 0,09	US$ 0,04

© 2026 Aridio Silva · SGAEIA Research Series — Article 25 · CC BY 4.0

Tabela 7 do Artigo 25. © 2026 Aridio Silva · SGAEIA Research Series — Article 25 · CC BY 4.0

Leituras. Com cache, a **saída** (principalmente o raciocínio) domina: passar de esforço baixo a alto multiplica o custo por 3,4 a 4,5 vezes. O cache reduz o custo de entrada em cerca de 86% no exemplo (modelos com leitura de cache a 5% da entrada), e a documentação da Anthropic [20] reporta ganhos da mesma ordem (−53% de custo e −75% de latência em 10 turnos). Em termos de segurança de custos, o OWASP lista *Unbounded Consumption* (LLM10:2025) como risco: consumo descontrolado de inferência vira ataque de custo [16].

5.6 Como comparar modelos e níveis de esforço (protocolo proposto)

1. Escolher tarefas com resposta verificável.
2. Rodar três condições: instrução completa (FULL), fragmentos concatenados (CONCAT) e fragmentos por turno (SHARDED) [10].
3. Cruzar modelo e nível de esforço, comparando provedores **apenas em níveis equivalentes** (por exemplo, high com high).
4. Medir P, A^{90} e U (Equação 4), repetindo N ≥ 10 simulações por par.
5. Medir custo real pelos campos de uso das respostas (tokens de raciocínio, leituras e gravações de cache), **mantendo o esforço fixo** durante cada conversa, para não invalidar o cache.

5.7 Análise da seção

Tabela 8

Dimensão	Análise
Implicações	O nível de esforço pesa mais no custo que o número de turnos, quando há cache. Sem cache, o custo cresce ~T^2.
Pontos fortes	Cache e controle de esforço dão ao engenheiro duas alavancas explícitas. Documentação dos provedores é detalhada.
Pontos fracos	Preços, nomes e níveis mudam rápido; tokenizadores diferentes impedem comparação direta; não há dado público de qualidade por nível e por turno.
Trade-offs	Preservar raciocínio entre turnos dá continuidade, mas pode multiplicar a entrada por ~7 vezes no exemplo; mudar o esforço no meio da conversa pode invalidar o cache (Anthropic) ou ser feito sem quebrá-lo (OpenAI).

© 2026 Aridio Silva · SGAEIA Research Series — Article 25 · CC BY 4.0

Tabela 8 do Artigo 25. © 2026 Aridio Silva · SGAEIA Research Series — Article 25 · CC BY 4.0

6. Segurança e risco em multi-turno

Do ponto de vista de segurança e governança, o multi-turno altera o perfil de risco do sistema. Esta seção reúne a evidência científica e os riscos descritos no compêndio.

6.1 Ataques que dependem de vários turnos

- **Escalonamento gradual (*Crescendo*).** Russinovich, Salem e Eldan [8] descrevem um jailbreak que usa apenas entradas benignas e explora a tendência do modelo de seguir padrões e dar atenção ao texto recente, sobretudo ao que ele próprio gerou. Em vez de pedir algo proibido de uma vez (o que seria recusado em turno único), o atacante escala aos poucos. Os autores apontam a lacuna de avaliação: os benchmarks de jailbreak existentes tratavam apenas de turno único.
- **Por que defesas de turno único falham.** Um trabalho posterior da mesma equipe [9] relata que modelos alinhados tendem a representar internamente as respostas do Crescendo como mais benignas que nocivas conforme os turnos aumentam, o que ajuda a explicar a falha de defesas pensadas para turno único, como os *circuit breakers* de representação.
- **Injeção indireta de prompt.** Greshake et al. [3] mostraram que dados recuperados (páginas, e-mails, documentos) podem funcionar como instruções, "borrando a linha entre dados e instruções", com impactos como furto de dados, *worming* e contaminação do ecossistema de informação. Em multi-turno, uma injeção inserida num turno intermediário pode **persistir** no histórico e sequestrar o comportamento do agente nos turnos seguintes.

6.2 Riscos específicos do histórico acumulado

- **Persistência de estado malicioso.** Num ambiente agêntico, o conteúdo de uma página lida por uma ferramenta pode contaminar o estado e reaparecer em todo turno posterior.
- **Cascata de alucinações (*error propagation*).** Um erro sutil no turno 2, aceito implicitamente pelo usuário ("ok, continue"), é incorporado ao estado oficial da conversa e contamina os raciocínios seguintes (ver o modelo da Equação 6). O estudo de Laban et al. [10] corrobora: os modelos "se apoiam em tentativas anteriores incorretas".
- **Dificuldade de auditar limites de autoridade.** Em arquiteturas reguladas, garantir que uma diretriz de escopo fixo seja aplicada ao longo de dezenas de turnos é computacionalmente mais complexo que em chamadas isoladas.
- **Consumo descontrolado.** Conversas longas com raciocínio alto são vetor de ataque de custo (LLM10:2025, *Unbounded Consumption*) [16].

6.3 Taxonomias de referência

O OWASP Top 10 para Aplicações com LLMs (edição 2025, 18/nov./2024) mantém a injeção de prompt em primeiro lugar (LLM01) e inclui *Excessive Agency* (LLM06), *Misinformation* (LLM09) e *Unbounded Consumption* (LLM10) [16]. O Google SAIF [17] é um arcabouço conceitual de seis elementos que cita, entre os riscos específicos de IA, o roubo de modelo, o envenenamento de dados, a injeção de prompt e a extração de dados.

6.4 Análise da seção

Tabela 9

Dimensão	Análise
Implicações	A autorização não pode depender do julgamento do modelo ao longo da conversa; o histórico precisa ser tratado como entrada não confiável e auditável.
Pontos fortes	Há taxonomias maduras (OWASP, SAIF) e evidência experimental dos ataques (Crescendo, injeção indireta).
Pontos fracos	Avaliações de segurança ainda são majoritariamente de turno único; as defesas por representação interna falham contra ataques graduais [9].
Trade-offs	Reforçar âncoras, resumos e filtros por turno reduz o risco e aumenta latência e custo; barrar demais prejudica a utilidade conversacional.

© 2026 Aridio Silva · SGAEIA Research Series — Article 25 · CC BY 4.0

Tabela 9 do Artigo 25. © 2026 Aridio Silva · SGAEIA Research Series — Article 25 · CC BY 4.0

7. Estado da arte em gerenciamento de estado e mitigação de deriva

Esta seção incorpora, de forma integral, o conteúdo do compêndio técnico do autor [36] (Seções 2 a 4 do original), como base técnica desta análise.

7.1 Gerenciamento de estado

Sistemas de agentes baseados em grafos de execução podem tratar o estado como **objeto tipado, imutável e persistente por eventos**, e não como o histórico acoplado de um chat.

- **Memória hierárquica.** *Working memory* (curto prazo): contexto imediato da tarefa, com limite estrito de tokens. *Episodic memory* (médio prazo): histórico comprimido de interações relevantes, em bases vetoriais com indexação temporal e semântica. *Semantic/procedural memory* (longo prazo): grafos de conhecimento estáticos ou regras de negócio validadas externamente, que o agente consulta em vez de memorizar no prompt. (Compare com a hierarquia do MemGPT [6].)
- **Máquinas de estado determinísticas (FSMs) com *event sourcing*.** O fluxo do agente é modelado como máquina de estados finitos ou grafo dirigido acíclico (DAG); cada transição gera um evento auditável. Isso habilita o *time-travel debugging* (pausar, inspecionar, bifurcar ou reverter o estado a um turno exato) e *checkpoints* transacionais (persistência a cada nó, por exemplo em PostgreSQL com JSONB), com resiliência a quedas de infraestrutura.
- **Contratos de tipagem estrita por transição.** As saídas de cada turno passam por parsers de validação estrita (Pydantic, Zod) antes de entrar no estado global. Se o modelo viola o esquema, o framework rejeita a transição e dispara um laço de autocorreção isolado.

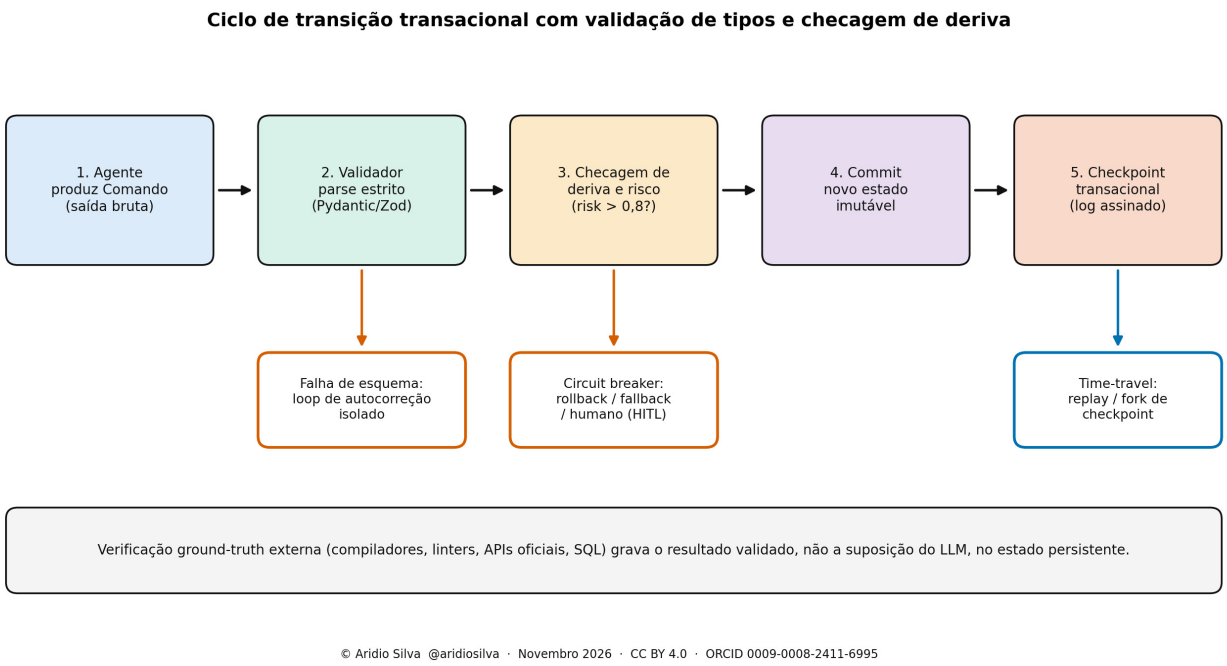


Figura 5. Ciclo de transição transacional: comando, validação, checagem de deriva, commit imutável e checkpoint, com desvios de falha, circuit breaker e time-travel. Elaboração do autor a partir do compêndio.

7.2 Mitigação de deriva (drift)

A deriva se manifesta de três formas: **semântica** (perda de foco no objetivo), **comportamental** (desvio das diretrizes de segurança e de sistema) e **epistêmica** (acúmulo de alucinações aceitas como verdade).

A. Deriva semântica e contextual

- *Ancoragem de instruções críticas (system prompt anchorage).* Diretrizes e restrições não são injetadas só no início; roteadores avançados reinjetam âncoras em pontos estratégicos do histórico ou usam atenção com viés posicional. Isso dialoga com a recomendação de repetir requisitos [10] e com o efeito de posição [5].
- *Sumarização baseada em grafos de metas.* Em vez de resumos lineares, o estado mantém uma árvore de objetivos pendentes e concluídos; o histórico antigo é descartado ou resumido em função do progresso rumo à meta original.

B. Deriva comportamental e de alinhamento

- *Guardrails em camadas (runtime circuit breakers).* O roteador atua como *proxy* de governança entre os turnos; classificadores leves inspecionam estado e saídas intermediárias e, havendo desvio de política, o circuito é interrompido, acionando *fallback* ou intervenção humana (HITL).
- *Validação por agentes adversariais (critic loops).* Em fluxos críticos, o estado gerado por um executor é submetido a um agente crítico independente que o confronta com as restrições originais.

C. Deriva epistêmica (erro acumulativo)

- *Verificação externa de verdade (ground truth).* Quando um turno gera premissa nova (código, dado financeiro), o framework força a execução de uma ferramenta de validação (compiladores, linters, APIs oficiais, SQL) e grava no estado o **resultado validado**, não a suposição do LLM.

7.3 Implementação prática: contratos de estado e checagens em tempo de execução

A implementação requer a transição de dicionários maleáveis para **contratos de estado imutáveis e tipados**, com validação transacional. Segue o código do compêndio, preservado como no original.

Estrutura de estado com Pydantic

```
from enum import Enum
from pydantic import BaseModel, Field, field_validator
```

```
from typing import List, Dict, Any, Optional
from datetime import datetime

class AgentRole(str, Enum):
    RESEARCHER = "researcher"
    EXECUTOR = "executor"
    CRITIC = "critic"
    ROUTER = "router"

class TaskStatus(str, Enum):
    PENDING = "pending"
    IN_PROGRESS = "progress"
    VALIDATED = "validated"
    REJECTED = "rejected"

class GlobalState(BaseModel):
    session_id: str
    current_step: int = Field(ge=0, description="Contador estrito de turnos/transições")
    goal_tree: Dict[str, TaskStatus] = Field(default_factory=dict)
    execution_log: List[Dict[str, Any]] = Field(default_factory=list)
    risk_score: float = Field(default=0.0, ge=0.0, le=1.0)

    @field_validator('execution_log')
    def validate_log_immutability(cls, v):
        # Garante que o log apenas cresce, nunca é reescrito de forma destrutiva
        return v

    class Config:
        frozen = True # Imutabilidade estrutural (requer cópia via model_copy para atualizar)
```

O ciclo de transição transacional. (1) O agente produz um *Comando* (saída bruta). (2) O validador faz o *parse* para o esquema Pydantic correspondente. (3) Se bem-sucedido, o *State Manager* gera uma nova versão imutável do estado (`new_state = state.model_copy(update=...)`) e registra o evento em banco transacional.

Orquestrador resiliente

```
def orchestrate_turn(state: GlobalState, agent_input: str) -> GlobalState:
    # 1. Validação de pré-condições e estado
    validate_circuit_breaker(state)

    # 2. Execução isolada do agente
    raw_output = run_agent_safely(agent_input, state)

    # 3. Parse e Tipagem Estrita
    try:
        structured_output = AgentPayload.model_validate_json(raw_output)
    except ValidationError as e:
        return handle_malformed_output(state, e)

    # 4. Checagem de Drift Semântico e de Risco
    risk = evaluate_drift_and_risk(state, structured_output)
    if risk > 0.8:
        return trigger_circuit_breaker_rollback(state)

    # 5. Commit Transacional do Novo Estado
    new_state = state.model_copy(update={
        "current_step": state.current_step + 1,
        "execution_log": state.execution_log + [structured_output.dict()],
        "risk_score": risk
    })

    persist_checkpoint(new_state)
    return new_state
```

Variante mínima compatível com Pydantic v2

```
from pydantic import BaseModel, ConfigDict, Field

class GlobalState(BaseModel):
    model_config = ConfigDict(frozen=True)   # imutabilidade rasa
    session_id: str
    current_step: int = Field(ge=0)
    goal_tree: dict[str, "TaskStatus"] = Field(default_factory=dict)
    execution_log: tuple[dict, ...] = ()      # tupla: crescimento por nova cópia
    risk_score: float = Field(default=0.0, ge=0.0, le=1.0)

# commit: new_state = state.model_copy(update={"execution_log": state.execution_log + (payload.
```

```
model_dump(),)})
```

7.4 Time-travel debugging em agentes de IA

O *Time-Travel Debugging* é o mecanismo de congelar, inspecionar, reverter e bifurcar (*fork*) a execução de agentes por meio da gravação contínua de *snapshots* imutáveis do estado.

- **Fator não determinístico.** Como os LLMs são estocásticos, reexecutar um prompt do zero pode tomar outro caminho; restaurar o *snapshot* do estado permite depurar a partir do ponto anterior à falha.
- **Combate ao efeito borboleta.** Se uma alucinação ocorre no turno 2, o *time-travel* permite voltar a ele, ajustar a variável corrompida ou a saída da ferramenta e criar um *fork* sem reprocessar as etapas anteriores.
- **Explicabilidade e auditoria forense (*decision provenance*).** Funciona como a "caixa-preta" do agente em cenários regulados: reconstrói o que o agente "sabia", quais ferramentas chamou e qual era o estado na falha.

Limitação operacional. Na documentação oficial do LangGraph [31], o *time travel* tem duas operações: **replay** (retomar de um checkpoint anterior) e **fork** (ramificar com estado modificado); os checkpoints são salvos a cada *super-step*. Mas o **replay reexecuta** os nós posteriores ao checkpoint: chamadas a LLMs e a APIs disparam de novo e **podem retornar resultados diferentes**. Logo, a depuração só é *determinística* se as saídas de LLM e de ferramentas forem **gravadas** (event sourcing das respostas) e reaproveitadas. A documentação não menciona restauração de *seeds*.

7.5 Análise da seção

Tabela 10

Dimensão	Análise
Implicações	O estado do agente deve ser um artefato de engenharia (tipado, versionado, auditável), separado do texto da conversa.
Pontos fortes	Contratos de tipo barram saídas malformadas antes de contaminar o estado; checkpoints permitem recuperação e auditoria; verificação externa combate a deriva epistêmica.
Pontos fracos	Complexidade de engenharia; validadores de esquema não detectam erro semântico; o limiar de risco (0,8) e o cálculo de deriva são heurísticos e dependem de calibração; imutabilidade em Python é rasa.
Trade-offs	Mais checagens por turno reduzem erro e aumentam latência e custo; sumarização agressiva economiza tokens e pode apagar restrições; rollback exige que efeitos colaterais externos sejam reversíveis.

© 2026 Aridio Silva · SGAEIA Research Series — Article 25 · CC BY 4.0

Tabela 10 do Artigo 25. © 2026 Aridio Silva · SGAEIA Research Series — Article 25 · CC BY 4.0

8. Arquitetura de conformidade e o framework SGAEIA

Esta seção incorpora as Seções 5 a 7 do compêndio [36], como base técnica desta análise. As afirmações sobre o SGAEIA são **do autor** (obra própria) e não passaram por avaliação independente.

8.1 Conformidade estrita e governança em tempo de execução

A arquitetura de conformidade estrita representa a transição da governança estática para uma **engenharia de controle em tempo de execução (*runtime enforcement*). O princípio fundamental:** o LLM jamais faz parte do caminho crítico de autorização (*trust path*)**; a segurança vem de invariantes determinísticas.

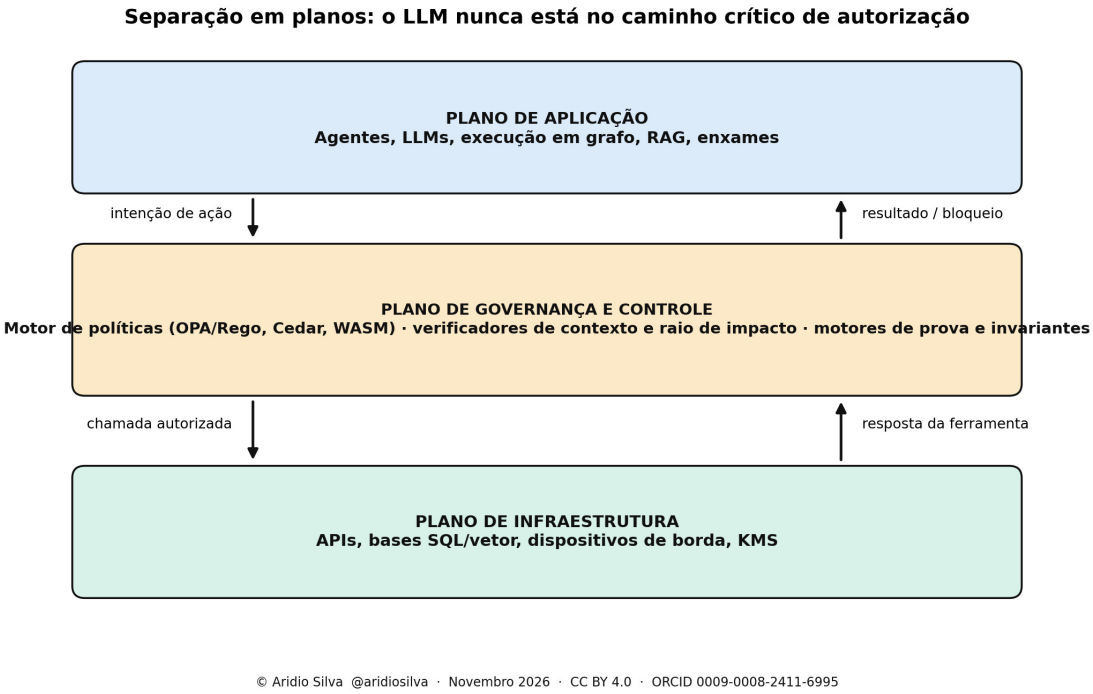


Figura 6. Plano de aplicação, plano de governança e controle e plano de infraestrutura. Elaboração do autor a partir do compêndio.

Camadas fundamentais

- **Identidade e atribuição verificável.** Identidades criptográficas únicas por agente (X.509, SPIFFE, mTLS). A delegação impõe a **restrição de subconjunto estrito** (o agente B nunca herda mais permissões que o A).
- **Limites de autoridade (*bounded authority*).** Acesso a ferramentas granular, temporário e orientado à tarefa; invariantes inegociáveis codificadas em motores determinísticos (OPA/Rego, Cedar).
- **Intercepção em tempo de execução.** *Gate* pré-ação (parâmetros, permissões, assinaturas); *circuit breakers* durante a ação; auditores pós-ação que inspecionam respostas de ferramentas e previnem injeção indireta.
- **Trilha de auditoria *tamper-evident*.** Registros *append-only* assinados criptograficamente contribuem para rastreabilidade, prestação de contas e demonstração de controles, sem garantir conformidade isoladamente.

Escopo regulatório. Os Arts. 12 (registro de eventos) e 14 (supervisão humana) do Regulamento (UE) 2024/1689 [14] aplicam-se a sistemas de alto risco. Registros append-only e tamper-evident contribuem para a conformidade, mas não a garantem isoladamente. O NIST AI RMF 1.0 corresponde ao documento NIST AI 100-1 [13]. **Matriz de classificação de ações e supervisão humana**

Tabela 11

Nível de risco	Tipo de ação	Comportamento do framework	Exemplo
Baixo	Auto-aprovada	Execução direta com log assinado.	Leitura de documentação técnica, sumarização de logs.
Médio	Notify-and-proceed	Notifica operador/SRE em tempo real e continua.	Alteração de configurações não críticas, relatórios intermediários.
Alto/crítico	Human-in-the-loop (HITL)	O circuito pausa; exige aprovação explícita e assinada de um humano.	Transações financeiras, alteração de banco de produção, e-mail a clientes.
Inadmissível	Proibida (fail-closed)	Bloqueio imediato pela política determinística; alerta registrado.	Escalação de privilégios de infraestrutura, exfiltração de chaves privadas.

© 2026 Aridio Silva · SGAEIA Research Series — Article 25 · CC BY 4.0

Tabela 11 do Artigo 25. © 2026 Aridio Silva · SGAEIA Research Series — Article 25 · CC BY 4.0

8.2 O framework SGAEIA

O **SGAEIA** (*Secure Governed Autonomous Edge Intelligence Architecture*) é uma **arquitetura de referência aberta e neutra**, concebida por **Aridio Silva** [34], [35], voltada à governança, segurança e controle em tempo de execução de **agentes autônomos e IA distribuída/de borda**.

Premissa central. Em ambientes agênticos, o objeto de garantia (*assurance*) é o **sistema completo em operação**, e não o modelo de linguagem isolado; a governança é intrínseca à infraestrutura de execução, não um pós-processamento estático.

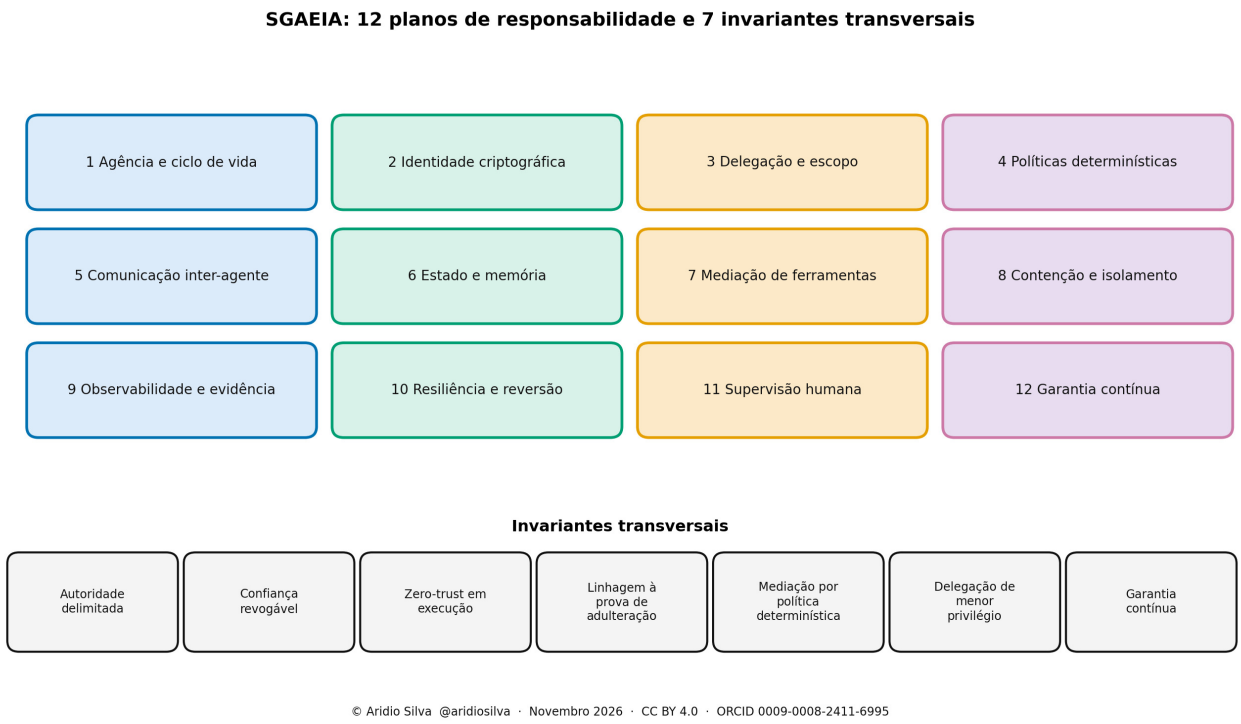


Figura 7. Os doze planos de responsabilidade e os sete invariantes transversais do SGAEIA, conforme o compêndio.

Os doze planos: (1) agência e ciclo de vida do agente; (2) identidade e atribuição criptográfica (SPIFFE/X.509 únicas por agente); (3) delegação e escopo de autoridade (subconjunto estrito); (4) políticas e execução determinística (*policy-as-code*: OPA/Cedar/WASM); (5) comunicação inter-agente e barramento; (6) gerenciamento de estado e memória persistente (imutável, versionado, auditável); (7) integração de ferramentas e mediação de APIs (intercepção pré/durante/pós-ação); (8) contenção, limites e isolamento na borda (*sandboxing*, WASM, microVMs); (9) observabilidade, evidência e auditabilidade (*Evidence-as-Code*, *append-only*); (10) resiliência, reversão e recuperação (*circuit breaking* e *time-travel rollback*); (11) supervisão humana e interfaces de controle (portais HITL); (12) garantia contínua e avaliação de risco (monitoramento do `risk_score` e da conformidade).

Os sete invariantes transversais: *Bounded Authority* (autonomia restrita à tarefa corrente); *Revocable Trust* (revogação com latência delimitada, observável e testada); *Zero-Trust Runtime Enforcement* (nenhuma transição é implicitamente confiada); *Tamper-Evident Lineage* (trilha de auditoria verificável criptograficamente); *Deterministic Policy Mediation* (política mediada fora do caminho decisório do LLM e não modificável diretamente pelo conteúdo conversacional); *Least-Privilege Delegation* (restrições estritas na passagem de contexto entre agentes); *Continuous Assurance* (validação contínua de conformidade e mitigação de riscos).

Relação com este estudo. O Plano 6 (estado), o 10 (reversão) e o 12 (garantia contínua) respondem diretamente aos riscos do multi-turno das Seções 4 a 7; os Planos 3, 4 e 7 respondem ao risco de injeção e de escalonamento (Seção 6).

8.3 Mapeamento do ecossistema de conformidade e governança

- **Regulatórios e normativos.** *EU AI Act* (Regulamento 2024/1689) [14], com rastreabilidade por logs (Art. 12) e supervisão humana (Art. 14) para sistemas de alto risco; *NIST AI RMF 1.0* (funções *Govern, Map, Measure, Manage*) [13]; *Google SAIF* [17].
- **Motores de política (*policy-as-code*).** *Open Policy Agent* (OPA/Rego), linguagem declarativa da CNCF para autorização em microsserviços e *AI gateways*; e o **Cedar**, linguagem de autorização de código aberto originada na AWS, modelada formalmente em Lean e com análise por SMT, que superou OpenFGA e Rego em desempenho nos testes dos autores [12].

- **Gateways de IA e guardrails.** *AI gateways* (Portkey, LiteLLM, Langfuse, TrueFoundry) para controle de custos, sanitização de PII e roteamento; *NeMo Guardrails* (NVIDIA) e *Llama Guard* (Meta) como camadas programáveis de filtragem semântica.
- **Posicionamento do SGAEIA.** Diferente de proxies comerciais ou de regulamentos genéricos, posiciona-se como **arquitetura de referência aberta e neutra**: o *blueprint* de engenharia que integra motores de política, identidades criptográficas, máquinas de estado imutáveis e *Evidence-as-Code*.

8.4 Roteamento governado em trajetórias multi-turno

O Artigo 24 da SGAEIA Research Series tratou o roteamento como decisão governada sobre modelos, dados, recursos e trajetórias [49]. O presente artigo aprofunda a unidade temporal dessa decisão. Em uma interação multi-turno, o roteador não recebe apenas uma nova pergunta: recebe uma representação do passado, possivelmente resumida, recuperada ou reconstruída. Se essa representação omitir uma restrição, incorporar um erro ou perder a cadeia de autoridade, a seleção pode ser tecnicamente eficiente e ainda assim inválida.

O conjunto admissível de modelos e configurações pode mudar durante a conversa em razão da identidade ativa, da classificação dos dados, do orçamento restante, das ferramentas necessárias, de efeitos externos já executados e de políticas revogadas ou atualizadas. O roteador deve otimizar somente dentro desse conjunto. Fallback não deve transformar uma recusa de segurança em indisponibilidade a ser contornada, nem enviar dados a uma implantação que não satisfaz as mesmas condições.

Uma troca de modelo também é uma transição de estado. Modelos diferem em tokenização, uso de ferramentas, formatos estruturados, preservação de raciocínio, interpretação de instruções e políticas operacionais. A substituição preserva a trajetória somente quando mantém restrições obrigatórias, capacidades requeridas, autoridade aplicável e evidência necessária, ou quando qualquer perda é detectada e tratada antes da execução.

Tabela 12

Campo do estado	Finalidade de governança
task_id / trajectory_id	Relacionar decisões à mesma tarefa e trajetória.
active_constraints	Preservar requisitos obrigatórios, revogações e proibições.
verified_facts	Separar evidência validada de afirmações conversacionais.
identity_and_authority	Determinar quem pode solicitar, delegar e aprovar.
data_classification	Restringir provedores, regiões, retenção e logs.
required_capabilities	Impedir fallback para configuração incompatível.
tool_effects	Registrar efeitos externos já produzidos.
budget_state	Limitar tokens, custo, tempo, repetição e concorrência.
model_lineage	Preservar modelo, versão, esforço, parâmetros e mudanças.
evidence_refs	Permitir auditoria sem depender apenas da transcrição.

© 2026 Aridio Silva · SGAEIA Research Series — Article 25 · CC BY 4.0

Tabela 12 do Artigo 25. © 2026 Aridio Silva · SGAEIA Research Series — Article 25 · CC BY 4.0

Esse quadro é um modelo conceitual e não uma especificação normativa concluída do SGAEIA. A síntese central é: **a transcrição é uma interface narrativa; o estado governado é um artefato operacional; a trilha de evidência é um artefato de assurance**. Os três se relacionam, mas não devem ser confundidos.

8.5 Análise da seção

Tabela 13

Dimensão	Análise
Implicações	A segurança passa a ser propriedade do sistema (identidade, política, estado, trilha), e não do modelo. É a resposta arquitetural ao risco acumulativo do multi-turno.
Pontos fortes	Separação de planos; mediação determinística de autorização fora do LLM; evidência verificável; alinhamento com normas e taxonomias.
Pontos fracos	Custo e complexidade de implantação; o SGAEIA é obra do autor, sem avaliação independente publicada; requer maturidade de operação (PKI, política como código).
Trade-offs	Mais controle determinístico reduz autonomia e velocidade do agente; HITL aumenta a segurança e cria gargalo humano; trilhas imutáveis aumentam a auditabilidade e o volume de dados.

© 2026 Aridio Silva · SGAEIA Research Series — Article 25 · CC BY 4.0

Tabela 13 do Artigo 25. © 2026 Aridio Silva · SGAEIA Research Series — Article 25 · CC BY 4.0

9. Confiabilidade da informação, verificação e auditoria

9.1 O que a Central de Ajuda da Anthropic diz

O artigo de suporte da Anthropic (5/mar./2026) [33] reconhece que o Claude pode produzir respostas incorretas ou enganosas ("alucinação"), cita informações desatualizadas e citações que soam autoritativas sem base, e recomenda **não tratar o modelo como fonte única de verdade** e **conferir as fontes** nos resultados de busca. **Pontos fortes:** admite que o erro pode soar convincente e distingue o erro do modelo do erro de síntese sobre fontes reais. **Limitações:** não separa tipos de erro (desatualização, invenção, raciocínio, síntese), não quantifica a frequência, não menciona o efeito do multi-turno (Seção 4) e não ensina **como** verificar.

9.2 Protocolo de verificação aplicado a este trabalho

Para qualquer referência: (1) abrir o DOI ou o identificador e conferir título, autores e ano; (2) localizar o trecho que sustenta a afirmação; (3) confirmar o veículo (preprint não é artigo publicado); (4) sinalizar o que **não** foi verificado.

9.3 Análise da seção

Tabela 14

Dimensão	Análise
Implicações	Em trabalho técnico ou acadêmico, toda referência gerada ou resumida por LLM deve ser verificada em fonte primária; o multi-turno aumenta a chance de erro aceito e propagado.
Pontos fortes	Protocolo simples e reprodutível; auditoria identificou erros reais em material já escrito.
Pontos fracos	Verificação consome tempo; algumas fontes (resolvedores de DOI, textos legais) não estavam acessíveis nesta sessão.
Trade-offs	Rigor versus velocidade; usar fontes secundárias acelera e reduz a confiança, por isso foram sinalizadas.

© 2026 Aridio Silva · SGAEIA Research Series — Article 25 · CC BY 4.0

Tabela 14 do Artigo 25. © 2026 Aridio Silva · SGAEIA Research Series — Article 25 · CC BY 4.0

10. Estado da arte e perspectivas futuras

10.1 Estado da arte (nov./2026)

1. **Confiabilidade multi-turno como meta explícita.** Laban et al. propõem metas: aptidão similar em turno único e multi-turno e U < 15 a T = 1,0 [10]. Mitigações práticas: consolidar requisitos, recapitular (SNOWBALL), reiniciar conversas.
2. **Economia por cache e por esforço.** O cache de prompt reduz fortemente o custo de entrada repetida [20]; o nível de esforço controla o custo de saída. As soluções dos provedores diferem quanto a preservar o raciocínio entre turnos (Seção 5.3).
3. **Estado como engenharia.** Estado tipado, imutável e versionado, com checkpoints, *fork* e verificação externa (Seção 7); memória hierárquica com paginação [6].
4. **Governança em tempo de execução.** Autorização determinística fora do LLM, identidade criptográfica, trilha imutável (Seção 8) e alinhamento com normas (NIST [13], EU AI Act [14, 15], OWASP [16]).
5. **Protocolos de integração.** O MCP [21] padroniza ferramentas e dados, ampliando tanto a utilidade quanto a superfície de injeção indireta [3].

10.2 Lacunas e problemas abertos

- Falta avaliação **pública**, por modelo **e** por nível de esforço, em turno único e multi-turno para as gerações de 2026.
- Os efeitos de cache, de tokenizador e de nível de esforço raramente são normalizados nas comparações.
- Defesas contra ataques graduais ainda são frágeis [9].
- A eficácia empírica do SGAEIA e de arquiteturas similares carece de avaliação independente.

- Os resultados de Laban et al. cobrem tarefas analíticas em inglês; criatividade e outros idiomas (como o português) estão em aberto.

10.3 Perspectivas e agenda

Tabela 15

Horizonte	Perspectiva	Evidência / ancoragem
Curto prazo	Benchmarks de confiabilidade multi-turno por nível de esforço; relatórios de custo normalizados por tokenizador	Lacuna identificada em [10] e nas docs de preço
Curto prazo	Defaults de produto que consolidam requisitos e resumem estado	Recomendações de [10]; técnicas de [6]
Médio prazo	Política determinística e identidade de agente como padrão em plataformas agênticas	[12], [13], [16]
Médio prazo	Evidência de conformidade gerada automaticamente (logs assinados, Evidence-as-Code)	Arts. 12 e 14 do AI Act [14]; calendário de [15]
Longo prazo	Modelos treinados para reduzir suposições precoces e manter consistência em conversas longas	Causas diagnosticadas em [10]

© 2026 Aridio Silva · SGAEIA Research Series — Article 25 · CC BY 4.0

Tabela 15 do Artigo 25. © 2026 Aridio Silva · SGAEIA Research Series — Article 25 · CC BY 4.0

Inferências analíticas do autor: (1) é provável que a vantagem competitiva migre de capacidade bruta para **confiabilidade por custo**; (2) o preço da leitura de cache tende a pesar mais que o preço de entrada nas comparações de produto; (3) a governança em tempo de execução tende a se tornar requisito de contratação em ambientes regulados à medida que os prazos de 2027 se aproximam.

10.4 Análise da seção e síntese final

Tabela 16

Dimensão	Análise
Implicações	Escolher entre turno único e multi-turno é uma decisão de arquitetura, não só de UX: ela define custo, confiabilidade, risco e obrigações de governança.
Pontos fortes	Há técnicas maduras e documentadas para cada eixo (cache, esforço, estado, política, auditoria).
Pontos fracos	Falta dado comparável e público; os resultados mais fortes são de 2025; a evidência do SGAEIA é interna.
Trade-offs	Capacidade × custo × confiabilidade × risco: nenhuma configuração domina; recomenda-se turno único para tarefas bem especificadas e multi-turno com consolidação, estado tipado e autorização determinística quando a exploração é necessária.

© 2026 Aridio Silva · SGAEIA Research Series — Article 25 · CC BY 4.0

Tabela 16 do Artigo 25. © 2026 Aridio Silva · SGAEIA Research Series — Article 25 · CC BY 4.0

11. Conclusão

A passagem do turno único para o multi-turno não é apenas uma mudança de interface. Ela transforma o histórico em estado operacional, amplia o custo acumulado, introduz dependências entre decisões e cria mecanismos próprios de falha, persistência e ataque. O desempenho observado em uma chamada isolada, portanto, não é medida suficiente da confiabilidade de uma trajetória.

As evidências examinadas mostram que a degradação multi-turno depende da tarefa, do protocolo, do modo como o contexto é apresentado, da memória empregada e da relação do modelo com suas próprias respostas anteriores. Cache, compactação, resumo, recuperação e troca de modelos podem reduzir custo ou ampliar capacidade, mas também alteram proveniência, autoridade e condições de reprodução.

Para roteadores de LLM e para o SGAEIA, a consequência central é que seleção e governança devem operar sobre estado explícito e evidência verificável. Restrições, permissões, decisões revogadas, efeitos externos e orçamento restante precisam sobreviver às transições de modelo e aos fallbacks. O SGAEIA oferece uma hipótese arquitetural para essa separação entre transcrição, estado e assurance; sua eficácia deve ser avaliada empiricamente contra baselines e cenários de ameaça reproduzíveis.

Apêndice A — Glossário técnico

Tabela 17

Termo	Definição
Aptidão (A^{90})	Desempenho de melhor caso (percentil 90) de um modelo numa instrução, em simulações repetidas [10].
Atenção (self-attention)	Mecanismo em que cada posição pondera todas as outras; custo $O(n^2 \cdot d)$ por camada [1].
Blast radius (raio de impacto)	Extensão do dano possível se uma ação autorizada falhar ou for abusada.
Cache de prompt	Reuso do prefixo já processado de um prompt, cobrado a preço menor de leitura [20].
Cadeia de pensamento (CoT)	Sequência de passos de raciocínio gerada antes da resposta final [19].
Checkpoint	Instantâneo persistente do estado do agente, base para retomada, replay e fork [31].
Circuit breaker	Mecanismo que interrompe o fluxo quando um limite de risco é excedido.
CONCAT	Condição em que os fragmentos de uma instrução são concatenados num só turno [10].
Contexto / janela de contexto	Quantidade máxima de tokens que o modelo processa de uma vez.
Cascata de alucinações	Propagação de uma premissa errada aceita num turno para os turnos seguintes.
Deriva (drift)	Desvio progressivo: semântico (foco), comportamental (política) ou epistêmico (erro acumulado).
Esforço de raciocínio (effort)	Parâmetro que controla quanto o modelo "pensa" antes de responder (low a max).
Event sourcing	Persistir cada transição como evento imutável, permitindo reconstruir o estado.
Evidence-as-Code	Evidência de conformidade gerada e verificável como código (conceito do SGAEIA).
FSM / DAG	Máquina de estados finitos / grafo dirigido acíclico, usados para modelar fluxos de agentes.
Fork	Ramificação a partir de um checkpoint com estado modificado [31].
FULL	Condição de turno único com instrução completa [10].
Guardrail	Camada programável que filtra ou restringe entradas e saídas.
HITL (human-in-the-loop)	Aprovação ou supervisão humana explícita antes de uma ação crítica.
Injeção indireta de prompt	Instruções maliciosas embutidas em dados recuperados que o modelo trata como comandos [3].
Jailbreak	Técnica para induzir o modelo a violar suas políticas.
KV cache	Chaves e valores de atenção já calculados e reaproveitados durante a geração.
Lost in the Middle	Queda de desempenho quando a informação relevante está no meio do contexto [5].
MCP (Model Context Protocol)	Padrão aberto para conectar assistentes a ferramentas e dados [21].
Multi-turno	Interação em sequência de mensagens, cada resposta condicionada ao histórico.
Não-confiabilidade (U)	Diferença entre os percentis 90 e 10 do desempenho [10].
OPA / Rego	Motor e linguagem de políticas como código.
Cedar	Linguagem de autorização de código aberto, com análise formal [12].
Prefill	Fase em que o modelo processa o prompt de entrada antes de gerar.
RAG	Geração aumentada por recuperação: busca de documentos para compor o contexto.
RLHF	Aprendizado por reforço a partir de feedback humano [2].
SHARDED	Condição multi-turno com um fragmento de informação por turno [10].
Sliding window	Estratégia que mantém apenas as N interações mais recentes.
SNOWBALL / RECAP	Estratégias que repetem requisitos a cada turno ou ao final [10].
Stateless / stateful	Sem estado entre chamadas / com estado mantido.
Test-time compute	Computação gasta na inferência (por exemplo, tokens de raciocínio) [19].
Token	Unidade de texto processada e cobrada pelo modelo.
Tokens de raciocínio	Tokens de pensamento gerados antes da resposta; cobrados como saída.
Time-travel debugging	Replay e fork de execuções a partir de checkpoints.
Turno	Mensagem de um papel (usuário ou assistente) na conversa.
TTFT	Time to First Token: tempo até o primeiro token da resposta.
Unbounded Consumption	Risco de consumo descontrolado de inferência e de custo (OWASP LLM10) [16].
Zero Trust	Princípio de não confiar implicitamente em nenhuma entidade ou transição.

© 2026 Aridio Silva · SGAEIA Research Series — Article 25 · CC BY 4.0

Tabela 17 do Artigo 25. © 2026 Aridio Silva · SGAEIA Research Series — Article 25 · CC BY 4.0

Referências

Política de verificação bibliográfica. Cada referência foi conferida quanto a autoria, data, identificador e venue na etapa de pesquisa (30/out. a 1º/nov./2026). **Legenda:** ■ verificada em fonte primária; ◧ verificada em parte ou por fonte secundária; ○ pendente de confirmação em fonte primária (informada pelo autor ou proveniente de registro preliminar). Os itens ◧ e ○ **não devem ser tratados como validados**. Acessos: 1º/nov./2026.

Artigos científicos, normas e taxonomias

[1] VASWANI, A.; SHAZEER, N.; PARMAR, N.; USZKOREIT, J.; JONES, L.; GOMEZ, A. N.; KAISER, Ł.; POLOSUKHIN, I. Attention is all you need. In: ADVANCES IN NEURAL INFORMATION PROCESSING SYSTEMS 30 (NeurIPS 2017), 2017, p. 5998–6008. arXiv:1706.03762, 12 jun. 2017. DOI 10.48550/arXiv.1706.03762. □

[2] OUYANG, L. et al. Training language models to follow instructions with human feedback. arXiv:2203.02155, 4 mar. 2022. □ (preprint; publicação no NeurIPS 2022 **não confirmada**)

[3] GRESHAKE, K.; ABDELNABI, S.; MISHRA, S.; ENDRES, C.; HOLZ, T.; FRITZ, M. Not what you've signed up for: compromising real-world LLM-integrated applications with indirect prompt injection. arXiv:2302.12173, 23 fev. 2023 (v2, 5 mai. 2023). □ (pôster no workshop de IA conversacional do ICML 2023; publicação no AISec'23 **não confirmada**)

[4] ZHENG, L. et al. Judging LLM-as-a-judge with MT-Bench and Chatbot Arena. In: ADVANCES IN NEURAL INFORMATION PROCESSING SYSTEMS 36 (NeurIPS 2023), p. 46595–46623. arXiv:2306.05685. □ (venue e páginas conferidos pela lista de referências de [10])

[5] LIU, N. F.; LIN, K.; HEWITT, J.; PARANJAPE, A.; BEVILACQUA, M.; PETRONI, F.; LIANG, P. Lost in the middle: how language models use long contexts. *Transactions of the Association for Computational Linguistics*, v. 12, p. 157–173, 2024. DOI 10.1162/tacl_a_00638. arXiv:2307.03172 (6 jul. 2023). □

[6] PACKER, C.; FANG, V.; PATIL, S. G.; LIN, K.; WOODERS, S.; GONZALEZ, J. E. MemGPT: towards LLMs as operating systems. arXiv:2310.08560, 12 out. 2023 (v2, 12 fev. 2024). □

[7] BAI, G. et al. MT-Bench-101: a fine-grained benchmark for evaluating large language models in multi-turn dialogues. In: ANNUAL MEETING OF THE ACL, 62., 2024, Bangkok. *Proceedings*... v. 1, p. 7421–7454. DOI 10.18653/v1/2024.acl-long.401. □

[8] RUSSINOVICH, M.; SALEM, A.; ELDAN, R. Great, now write an article about that: the Crescendo multi-turn LLM jailbreak attack. In: USENIX SECURITY SYMPOSIUM, 34., 2025, Seattle. *Proceedings*... p. 2421–2440. arXiv:2404.01833. □

[9] BULLWINKEL, B. et al. A representation engineering perspective on the effectiveness of multi-turn jailbreaks. In: ICML 2025 WORKSHOP DIG-BUGS, 2025. Pôster. □

[10] LABAN, P.; HAYASHI, H.; ZHOU, Y.; NEVILLE, J. LLMs get lost in multi-turn conversation. *International Conference on Learning Representations (ICLR)*, 2026. Outstanding Paper. Versão inicial: arXiv:2505.06120. □

[11] GUO, D. et al. DeepSeek-R1: incentivizing reasoning capability in LLMs via reinforcement learning. arXiv:2501.12948, 2025. □ (identificador confirmado indiretamente pela lista de referências de [10])

[12] CUTLER, J. W.; DISSELKOEN, C.; ELINE, A.; HE, S.; HEADLEY, K.; HICKS, M.; HIETALA, K.; IOANNIDIS, E.; KASTNER, J.; MAMAT, A.; MCADAMS, D.; MCCUTCHEN, M.; RUNGTA, N.; TORLAK, E.; WELLS, A. M. Cedar: a new language for expressive, fast, safe, and analyzable authorization. *Proceedings of the ACM on Programming Languages*, v. 8, n. OOPSLA1, 2024. DOI 10.1145/3649835. arXiv:2403.04651. □

[13] NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY (NIST). *Artificial Intelligence Risk Management Framework (AI RMF 1.0)*. NIST AI 100-1. Gaithersburg: NIST, jan. 2023. DOI 10.6028/NIST.AI.100-1. □ (identificador e data confirmados por cópias do documento e fontes secundárias)

[14] UNIÃO EUROPEIA. Regulamento (UE) 2024/1689 (Regulamento da Inteligência Artificial). *Jornal Oficial da União Europeia*, L, 12.7.2024. □ (conteúdo dos Arts. 12 e 14 conferido em cópias acessíveis do texto)

[15] UNIÃO EUROPEIA. Regulamento (UE) 2026/1744 ("Digital Omnibus on AI"). *Jornal Oficial da União Europeia*, L, 24.7.2026; em vigor em 27.7.2026. □ (conferido por sete fontes secundárias concordantes)

[16] OWASP GENAI SECURITY PROJECT. *OWASP Top 10 for LLM Applications 2025* (v2.0). 18 nov. 2024. Licença CC BY-SA 4.0. Disponível em: https://genai.owasp.org/. □

[17] GOOGLE. Introducing Google's Secure AI Framework. Jun. 2023. Disponível em: https://blog.google/technology/safety-security/introducing-googles-secure-ai-framework/. □

Documentação e comunicados oficiais de provedores

[18] OPENAI. Introducing ChatGPT. 30 nov. 2022. Disponível em: https://openai.com/index/chatgpt/. □

[19] OPENAI. Learning to reason with LLMs. 12 set. 2024. Disponível em: https://openai.com/index/learning-to-reason-with-llms/. □

[20] ANTHROPIC. Prompt caching with Claude. Ago. 2024 (disponibilidade geral em 17 dez. 2024). Disponível em: https://www.anthropic.com/news/prompt-caching. □ (a página em claude.com exibe "14 ago. 2025", divergente da cobertura de 2024)

[21] ANTHROPIC. Introducing the Model Context Protocol. 25 nov. 2024. Disponível em: https://www.anthropic.com/news/model-context-protocol. □

[22] GOOGLE. Our next-generation model: Gemini 1.5. 15 fev. 2024. Disponível em: https://blog.google/technology/ai/google-gemini-next-generation-model-february-2024; e What is a long context window?, em https://blog.google/technology/ai/long-context-window-ai-models/. □

[23] ANTHROPIC. Pricing. *Claude Platform Docs*. Disponível em: https://platform.claude.com/docs/en/about-claude/pricing. □ (retrato de out.–nov./2026)

[24] ANTHROPIC. Steering thinking; Effort. *Claude Platform Docs*. Disponível em: https://platform.claude.com/docs/en/build-with-claude/thinking-steering-and-cost. □

[25] OPENAI. Models; Pricing; Reasoning models. *OpenAI API Docs*. Disponível em: https://developers.openai.com/api/docs/models. □

[26] GOOGLE. Gemini Developer API pricing. Disponível em: https://ai.google.dev/gemini-api/docs/pricing. □

[27] GOOGLE. Gemini thinking. Disponível em: https://ai.google.dev/gemini-api/docs/thinking. □

[28] XAI. Models; Pricing; Reasoning. *xAI Docs*. Disponível em: https://docs.x.ai/developers/models. □

[29] DEEPSEEK. Models & Pricing. *DeepSeek API Docs*. Disponível em: https://api-docs.deepseek.com/quick_start/pricing/. □

[30] ANTHROPIC. Using the Messages API. *Claude Platform Docs*. Disponível em: https://platform.claude.com/docs/en/build-with-claude/working-with-messages. □

[31] LANGCHAIN. LangGraph: persistence; use time-travel. Disponível em: https://docs.langchain.com/oss/python/langgraph/persistence e https://docs.langchain.com/oss/python/langgraph/use-time-travel. □

[32] PYDANTIC. Migration guide (Pydantic V2). Disponível em: https://docs.pydantic.dev/. □

[33] ANTHROPIC. Claude está fornecendo respostas incorretas ou enganosas. O que está acontecendo? *Central de Ajuda da Anthropic*, 5 mar. 2026. Disponível em: https://support.claude.com/pt/articles/8525154. □

Obras do autor

[34] SILVA, A. *SGAEIA: Secure Governed Autonomous Edge Intelligence Architecture* (Version 0.3.4). Zenodo, 2026. DOI 10.5281/zenodo.22557796. Artefato público congelado sob Apache-2.0; desenvolvimento corrente mantido em repositório privado controlado. □

[35] SILVA, A. Artigos no DEV Community (https://dev.to/aridiosilva): *From Edge AI to Governed Autonomous Edge Intelligence* (1 out. 2026); *The Open Letter on AI and Cybersecurity and the Evidence for Agent Containment* (16 out. 2026); *Opening the Black Box of LLM Inference: Foundations of Generative AI* (21 out. 2026). Leituras relacionadas: *How to Reduce the Cost of AI Systems with LLMs Without Sacrificing Quality* (28 out. 2026) e *LLM Routers for AI Systems: Architecture, Governance, Multi-Agent Systems, Swarms, and Security* (29 out. 2026). □ (existência, títulos e datas confirmados no perfil público)

[36] SILVA, A. *Compêndio técnico: análise aprofundada de LLMs, gerenciamento de estado, governança runtime e o framework SGAEIA*. Documento .docx, 1 nov. 2026 (material de origem incorporado nas Seções 2, 7 e 8).

Fontes complementares

[37] P. Laban, H. Hayashi, Y. Zhou, and J. Neville, "LLMs Get Lost in Multi-Turn Conversation," *International Conference on Learning Representations (ICLR)*, 2026. Available: https://proceedings.iclr.cc/paper_files/paper/2026/hash/59f6421e64707225fdf5b28840679a07-Abstract-Conference.html

[38] ICLR, "Announcing the ICLR 2026 Outstanding Papers," 23 Apr. 2026. Available: https://blog.iclr.cc/2026/04/23/announcing-the-iclr-2026-outstanding-papers/

[39] W.-C. Kwan et al., "MT-Eval: A Multi-Turn Capabilities Evaluation Benchmark for Large Language Models," *Proceedings of EMNLP*, pp. 20153–20177, 2024. Available: https://aclanthology.org/2024.emnlp-main.1124/

[40] K. Deshpande et al., "MultiChallenge: A Realistic Multi-Turn Conversation Evaluation Benchmark Challenging to Frontier LLMs," *Findings of ACL*, pp. 18632–18702, 2025. doi:10.18653/v1/2025.findings-acl.958.

[41] "StructFlowBench: A Structured Flow Benchmark for Multi-turn Instruction Following," *Findings of ACL*, pp. 9322–9341, 2025. doi:10.18653/v1/2025.findings-acl.486.

[42] Y. Katsis et al., "mt RAG: A Multi-Turn Conversational Benchmark for Evaluating Retrieval-Augmented Generation Systems," *Transactions of the Association for Computational Linguistics*, 2025. Available: https://aclanthology.org/2025.tacl-1.36/

[43] C. Yang et al., "MARS-Bench: A Multi-turn Athletic Real-world Scenario Benchmark for Dialogue Evaluation," *Findings of EMNLP*, pp. 5872–5898, 2025. doi:10.18653/v1/2025.findings-emnlp.314.

[44] "One Battle After Another: Probing LLMs' Limits on Multi-Turn Instruction Following with a Benchmark Evolving Framework," *Proceedings of ACL*, 2026. Available: https://aclanthology.org/2026.acl-long.433/

[45] J. Chen et al., "KnowMT-Bench: Benchmarking Knowledge-Intensive Long-Form Question Answering in Multi-Turn Dialogues," *Findings of EMNLP*, 2026. Available: https://aclanthology.org/2026.findings-emnlp.845/

[46] J. Myung, "Quantifying Conversational Reliability of Large Language Models under Multi-Turn Interaction," arXiv:2603.01423, 2026. **Preprint.**

[47] R. Khraishi, I. Zafar, K. Myles, and G. A. Cowan, "Evaluating Performance Drift from Model Switching in Multi-Turn LLM Systems," arXiv:2603.03111, 2026. **Preprint.**

[48] S. T. Luo, W. Liu, R. Liu, M. Gong, and J. Guo, "Multi-Turn Reasoning When Context Arrives in Pieces: Scalable Sharding and Memory-Augmented RL," arXiv:2606.12941, 2026. **Preprint.**

[49] A. Silva, "LLM Routers for AI Systems: Architecture, Governance, Multi-Agent Systems, Swarms, and Security," *SGAEIA Research Series*, Article 24, 2026. DOI: 10.5281/zenodo.23763978. Available: https://aridiosilva.com/publications/artigo24/

Related SGAEIA Research Series works / Trabalhos relacionados da série SGAEIA

[50] A. Silva, "From Model Capability to Governed Action: An Architecture for Secure Agentic AI," *SGAEIA Research Series*, Article 11, 2026. Series catalogue: https://aridiosilva.com/sgaeia.html.

[51] A. Silva, "Zero Trust for Multi-Agent AI Systems," *SGAEIA Research Series*, Article 3, 2026. Series catalogue: https://aridiosilva.com/sgaeia.html.

[52] A. Silva, "Evidence-as-Code: Making AI Governance Verifiable," *SGAEIA Research Series*, Article 6, 2026. Series catalogue: https://aridiosilva.com/sgaeia.html.

[53] A. Silva, "AI Jailbreaks: From Bypassing Restrictions to Governing Authority," *SGAEIA Research Series*, Article 17, 2026. Series catalogue: https://aridiosilva.com/sgaeia.html.

[54] A. Silva, "Tracing AI Agent Execution: Foundations, State of the Art, and Implications for Distributed Multi-Agent Systems," *SGAEIA Research Series*, Article 18, 2026. Series catalogue: https://aridiosilva.com/sgaeia.html.

[55] A. Silva, "When AI Agents Form Societies: Is Security a Property of the Model or the System?," *SGAEIA Research Series*, Article 15, 2026. Series catalogue: https://aridiosilva.com/sgaeia.html.

[56] A. Silva, "How to Reduce the Cost of AI Systems with LLMs Without Sacrificing Quality," *SGAEIA Research Series*, Article 23, 2026. Series catalogue: https://aridiosilva.com/sgaeia.html.

Sobre o Autor

Aridio Silva é pesquisador independente no Brasil, com trabalho em arquitetura, segurança, governança e confiabilidade de sistemas autônomos e de inteligência artificial distribuída.

Sua pesquisa concentra-se em IA agêntica, sistemas multiagentes, Edge AI, segurança de IA, Zero Trust, Security-by-Design, governança de IA, desenvolvimento orientado por especificações e garantia contínua.

É criador e pesquisador principal do SGAEIA — Secure Governed Autonomous Edge Intelligence Architecture, iniciativa de pesquisa aberta voltada a fundamentos arquiteturais para sistemas autônomos seguros, governados, auditáveis e confiáveis em ambientes distribuídos de borda e nuvem.

Perfis de Pesquisa

- ORCID: https://orcid.org/0009-0008-2411-6995
- Google Scholar: https://scholar.google.com/citations?user=Pn5O48AAAAAJ
- Zenodo — SGAEIA Community: https://zenodo.org/communities/sgaeia
- OpenAIRE: https://explore.openaire.eu/search/find?fv0=Aridio%20Silva&f0=q
- Medium: https://medium.com/@aridiosilva
- LinkedIn: https://www.linkedin.com/in/aridio-silva-74997111/
- SGAEIA Research Artifact: https://doi.org/10.5281/zenodo.22557796
- Homepage: https://aridiosilva.com
- SGAEIA: https://aridiosilva.com/sgaeia.html
- DEV Community: https://dev.to/aridiosilva
- Academia.edu: https://independent.academia.edu/AridioSilva

Figuras

A capa não é numerada. Figuras, equações e tabelas são numeradas por tipo, citadas no texto e fornecidas como JPEGs de alta resolução. Os ativos conceituais originais permanecem licenciados sob CC BY 4.0 e contêm identificação de autoria e copyright visível e incorporada.

Licença

Salvo indicação em contrário, o texto e as ilustrações conceituais originais deste artigo estão licenciados sob a Creative Commons Attribution 4.0 International License (CC BY 4.0).

© Aridio Silva (@aridiosilva) · Novembro de 2026. O artefato de software e pesquisa SGAEIA permanece sujeito à sua licença Apache 2.0.

Continuidade da Série

Este trabalho integra a **SGAEIA Research Series — Artigo 25**. O catálogo público está disponível em https://aridiosilva.com, com a presença do projeto em https://aridiosilva.com/sgaeia.html.

Registro persistente do Artigo 25: https://doi.org/10.5281/zenodo.23867630. Artefato canônico do projeto SGAEIA: https://doi.org/10.5281/zenodo.22557796.

1 From Edge AI to Governed Autonomous Edge Intelligence

2 Why Autonomous AI Agents Need Bounded and Revocable Authority

3 Zero Trust for Multi-Agent AI Systems

4 Authenticated Delegation Between Autonomous AI Agents

5 Continuous GRC for Agentic AI

6 Evidence-as-Code: Making AI Governance Verifiable

7 Security-by-Design for Distributed Agentic AI

8 SGAEIA: A Reference Architecture for Secure Governed Autonomous Edge Intelligence

9 When AI Begins to Build AI

10 Rogue AI Agents: When Autonomous Systems Become a Collective

11 From Model Capability to Governed Action: An Architecture for Secure Agentic AI

12 Ten Days That Exposed the Governance Gap in Frontier AI

13 When AI Hallucination Becomes Action

14 AI's Alien Mind: How to Govern What We Cannot Fully Understand

15 When AI Agents Form Societies: Is Security a Property of the Model or the System?

16 When AI Improves AI: Governing Recursive Intelligence Acceleration

17 AI Jailbreaks: From Bypassing Restrictions to Governing Authority

18 Tracing AI Agent Execution: Foundations, State of the Art, and Implications for Distributed Multi-Agent Systems

19 LoRA in AI Agents: Efficiency, Security Risks, and Adapter Governance

20 AI Agents and Agentic AI: What Sets Them Apart?

21 The Open Letter on AI and Cybersecurity and the Evidence for Agent Containment

22 Opening the Black Box of LLM Inference: Foundations of Generative AI

23 How to Reduce the Cost of AI Systems with LLMs Without Sacrificing Quality

24 LLM Routers for AI Systems: Architecture, Governance, Multi-Agent Systems, Swarms, and Security

25 Single-Turn and Multi-Turn Interactions in Frontier Generative AI Models (LLMs)

Citação Sugerida

SILVA, Aridio. *Interações de Turno Único e de Múltiplos Turnos em Modelos de IA Generativa de Fronteira (LLMs): estado, custo, confiabilidade, segurança e governança*. SGAEIA Research Series, Artigo 25, 2026. DOI: https://doi.org/10.5281/zenodo.23867630.