

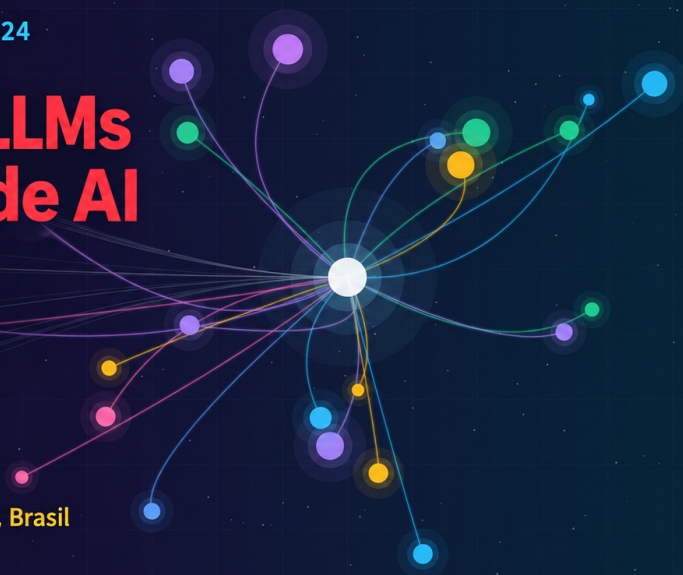
SÉRIE DE PESQUISA SGAEIA — ARTIGO 24

Roteadores de LLMs para Sistemas de AI

Arquitetura, Governança,
Multiagentes, Enxames
e Segurança

Aridio Silva — Pesquisador Independente, Brasil
ORCID 0009-0008-2411-6995

© Aridio Silva @aridiosilva — Outubro 2026



Roteadores de LLMs para Sistemas de AI

Arquitetura, governança, multiagentes, enxames e segurança

SGAEIA Research Series — Article 24

Aridio Silva — Independent Researcher, Brazil

ORCID 0009-0008-2411-6995

29 de outubro de 2026 · CC BY 4.0

DOI: 10.5281/zenodo.23763978

Roteadores de LLMs para Sistemas de AI: Arquitetura, Governança, Multiagentes, Enxames e Segurança

Da otimização de custo à decisão governada sobre modelos, dados e trajetórias

Série de Pesquisa SGAEIA — Artigo 24

Aridio Silva — Pesquisador Independente, Brasil

Criador do SGAEIA — Secure Governed Autonomous Edge Intelligence Architecture

ORCID: 0009-0008-2411-6995

29 de outubro de 2026

© 2026 Aridio Silva. Texto e ilustrações originais sob CC BY 4.0.

Resumo

O roteamento de modelos de linguagem permite distribuir requisições entre modelos, provedores e configurações com diferentes capacidades, custos e tempos de resposta. Sua utilidade, porém, depende de duas decisões distintas: determinar quais alternativas são admissíveis para a tarefa e escolher, entre elas, a configuração mais adequada. Este artigo apresenta uma revisão técnica do tema e sustenta que, em sistemas agênticos e multiagentes, o roteamento integra uma fronteira de decisão sobre autoridade, dados, recursos e continuidade das políticas. A análise distingue roteador, gateway e plano de controle; compara regras, classificadores, similaridade, cascatas e seleção orientada por modelos; e examina o que os benchmarks permitem concluir sobre qualidade, custo, latência e robustez. A contribuição analítica consiste em um quadro de avaliação de roteamento governado ligado ao SGAEIA, aplicado a cenários de substituição de modelos, orçamento e execução multiagente. A análise mostra por que economia por chamada não equivale a economia por tarefa, disponibilidade não assegura substituição admissível e acurácia média não demonstra integridade de uma trajetória. A adoção deve depender de ganhos observados contra referências simples, acompanhados de evidência suficiente sobre os limites e as consequências das decisões.

Palavras-chave: AI; roteamento de LLMs; AI gateway; sistemas multiagentes; segurança; governança; avaliação; SGAEIA.

Sumário

1. O problema vai além de escolher o modelo mais barato
2. Roteador, gateway e plano de controle
3. Como a evidência foi organizada
4. Estratégias de roteamento e seus compromissos
5. Qualidade, custo e latência precisam da mesma unidade de análise
6. Integração e escolha de infraestrutura
7. O que os benchmarks demonstram
8. Comparação transversal das estratégias
9. Identidade, autoridade e substituição governada
10. Agentes, multiagentes e enxames
11. Governança, segurança e evidências
12. Quadro SGAEIA de análise de roteamento governado
13. Evolução e maturidade devem ser avaliadas por dimensão
14. Quando adotar e quando manter uma solução mais simples
15. Discussão e limites da análise
16. Conclusão

1. O problema vai além de escolher o modelo mais barato

Uma aplicação que usa modelos de linguagem precisa decidir onde investir capacidade de inferência. Um modelo adequado para extrair campos de um documento pode ser insuficiente para verificar uma solução complexa; um modelo competente nessa verificação pode apresentar custo ou tempo de resposta incompatível com tarefas repetitivas. A literatura de roteamento explora essa heterogeneidade para distribuir consultas entre alternativas, em vez de manter uma escolha fixa para todo o tráfego. O potencial econômico nasce da complementaridade dos modelos e da diversidade das tarefas, mas depende da capacidade de antecipar ou reconhecer onde cada alternativa funciona [1][5][6].

O problema muda quando a resposta produzida participa de uma ação. Num assistente de consulta, a seleção inadequada pode gerar uma resposta insatisfatória; num agente que consulta sistemas, escreve código ou delega trabalho, a mesma decisão pode alterar dados transmitidos, ferramentas utilizadas e efeitos posteriores. A escolha do modelo passa a depender do estado da tarefa, de quem autorizou a execução e dos limites que devem acompanhar essa autorização. O SGAEIA oferece a perspectiva arquitetural adotada neste artigo: capacidade de produzir uma resposta e autoridade para conduzir uma ação são propriedades diferentes [27].

A tese central é que o roteamento em sistemas agênticos deixa de ser apenas uma otimização de inferência e integra uma fronteira crítica de decisão, governança e produção de evidências. Isso não transforma todo roteador em um sistema de segurança nem torna obrigatório usar um algoritmo aprendido. O estudo de Shafran e colaboradores já situa roteadores entre os planos de controle de LLMs e demonstra ataques dirigidos à seleção do modelo [9]. O passo desenvolvido aqui é organizar as implicações desse entendimento em critérios comparáveis, decisões de engenharia e evidências verificáveis.

Seis perguntas orientam a análise: em quais condições o roteamento economiza preservando a qualidade; quando uma estratégia supera alternativas simples; que valor o gateway oferece independentemente do algoritmo; quais riscos decorrem da concentração de credenciais e políticas; o que muda em agentes e

enxames; e que evidência permite governar seleção, substituição e revogação. A contribuição do artigo está na síntese dessas perguntas em um mesmo quadro de decisão, apresentado e aplicado na Seção 12. Seu valor é analítico: permitir examinar uma configuração, identificar seus pressupostos e explicar uma decisão de adoção ou rejeição. A eficácia de uma implementação concreta exige avaliação no ambiente em que será utilizada.

2. Roteador, gateway e plano de controle

Um roteador de LLMs é uma função de seleção: recebe informações sobre uma tarefa e escolhe um modelo, uma implantação ou uma configuração de execução. Essa função pode consistir em regras locais, um classificador, um cálculo de similaridade ou uma política que usa resultados intermediários. Sua existência não pressupõe um servidor HTTP ou uma API pública; a seleção pode ocorrer dentro do próprio processo da aplicação. A representação como decisão sequencial, adotada pelo LLMRouter, ajuda a compreender casos que incluem histórico, personalização e múltiplas etapas [8].

Um AI gateway medeia o acesso aos modelos. Pode adaptar protocolos, verificar credenciais, aplicar limites, encaminhar requisições e produzir telemetria, além de hospedar uma função de roteamento. Essas responsabilidades continuam úteis mesmo quando todas as consultas seguem para um único modelo. A documentação de chaves virtuais do LiteLLM, por exemplo, descreve controle de acesso a modelos e acompanhamento de gastos; isso caracteriza uma capacidade de mediação, não uma demonstração de que seu seletor obtenha a melhor relação entre custo e qualidade [10].

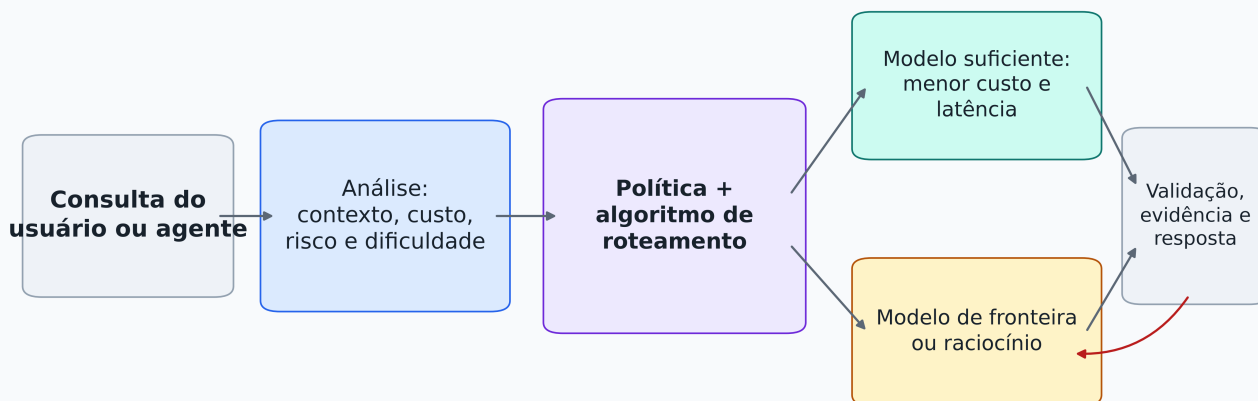
O plano de controle administra as decisões que governam o ambiente: políticas, configurações, permissões, versões e critérios de substituição. O plano de dados executa o tráfego correspondente. Um gateway participa da aplicação dessas decisões, mas seus componentes administrativos, seu caminho de inferência e seu armazenamento de evidências não precisam compartilhar o mesmo domínio de privilégios. A distinção tem consequências práticas: comprometer uma aplicação cliente não deveria permitir editar a política do gateway, e observar uma resposta não deveria conceder acesso às credenciais que a tornaram possível [16].

Tabela 1

Elemento	Pergunta principal	Responsabilidade	Evidência de avaliação
Roteador	Qual alternativa deve atender esta tarefa?	Selecionar modelo, configuração ou próxima etapa	Qualidade da escolha, custo de decisão e generalização
Gateway	Como mediar a execução autorizada?	Adaptar acesso, aplicar controles e observar o tráfego	Isolamento, disponibilidade, limites e registros
Plano de controle	Quem pode definir e alterar essas decisões?	Administrar políticas, permissões e ciclo de vida	Autorização das mudanças, versões e possibilidade de revogação

Figura 1. Decisão governada de roteamento

A consulta passa por política e classificação antes da seleção do modelo; a validação pode escalar ou bloquear.



Objetivo: selecionar capacidade suficiente sem romper política, autoridade ou evidência.

Fonte: Ong et al. (2025); Jiangeng et al. (2026); SGAEIA interpretation

© 2026 Aridio Silva | Project SGAEIA | CC BY 4.0

Figura 1. O fluxo distingue análise da tarefa, decisão de roteamento e validação do resultado. A seleção deve considerar as restrições aplicáveis antes de otimizar desempenho. O desenho é conceitual: a presença desses blocos não comprova que uma implantação aplique as políticas corretamente.

Confundir essas camadas produz dois erros de avaliação. O primeiro é atribuir ao algoritmo de seleção benefícios que vêm da infraestrutura, como inventário de chaves ou auditoria centralizada. O segundo é considerar que um gateway com muitos recursos já resolveu qualidade, autorização e segurança de agentes. Uma arquitetura consistente define a responsabilidade de cada componente e verifica suas interações, inclusive quando o componente está indisponível ou opera com informação incompleta.

3. Como a evidência foi organizada

A análise é uma revisão técnica narrativa, orientada pelas seis perguntas da introdução. O corpus de partida foi o material do Artigo 24, complementado pela consulta a trabalhos originais, registros editoriais, documentação oficial, referências de gestão de riscos e relatos técnicos de incidentes. A seleção privilegia fontes que explicam um mecanismo, apresentam avaliação comparável ou documentam uma condição operacional relevante. A data de referência desta revisão é 29 de outubro de 2026; ela não corresponde a uma busca exaustiva de toda a produção sobre roteamento.

Foram separadas quatro classes de afirmação. Resultados de métodos e benchmarks são atribuídos aos seus autores e às condições avaliadas. Recursos de produtos são descritos como funcionalidades documentadas pelos fornecedores, enquanto requisitos legais e de gestão de riscos permanecem vinculados às suas fontes oficiais. As comparações transversais, a matriz de governança e os cenários de decisão são sínteses analíticas desenvolvidas neste artigo. Essa separação permite distinguir o que foi observado por terceiros do que é uma consequência argumentada ou uma proposta de avaliação.

Um mesmo percentual de economia pode representar menos chamadas ao modelo caro, menor cobrança de API ou menor custo computacional. Por isso, a comparação registra a unidade, a referência usada e o limite de generalização, em vez de ordenar trabalhos pelos maiores percentuais. Também distingue estudos que propõem um método de trabalhos que reavaliam vários métodos sob um protocolo comum. Nenhuma dessas classes dispensa examinar dados, modelos, avaliadores e possíveis interesses comerciais.

O artigo não reapresenta os resultados publicados como experimentos próprios. Os cálculos da Seção 5 são cenários hipotéticos com entradas explícitas e aritmética reproduzível. A contribuição é demonstrada pela consistência da comparação e pela aplicação do quadro de governança a decisões em que critérios isolados

levam a conclusões insuficientes. Uma avaliação empírica de redução de risco, desempenho de controles ou segurança de produção pertence a uma etapa posterior, com protocolo e resultados próprios.

4. Estratégias de roteamento e seus compromissos

4.1 Escolher antes da geração

O roteamento preditivo escolhe o destino usando informações disponíveis antes da resposta: conteúdo da consulta, domínio, tamanho, identidade, histórico ou uma representação vetorial. Regras explícitas podem encaminhar documentos de determinada classe a uma implantação autorizada ou reservar um modelo a tarefas que exigem uma capacidade específica. Sua transparência facilita explicar a decisão e verificar exceções; sua limitação é a dificuldade de representar fronteiras complexas sem proliferar regras. Uma regra por palavra-chave, por exemplo, pode confundir uma pergunta sobre programação com uma solicitação que apenas menciona uma linguagem.

Classificadores e modelos de preferência substituem parte dessas regras por uma estimativa aprendida. O RouteLLM utiliza dados de preferências para estimar quando vale recorrer ao modelo mais capaz e investiga diferentes arquiteturas de seleção [1]. O Hybrid LLM também ajusta o encaminhamento entre alternativas menores e maiores segundo uma qualidade desejada [4]. A vantagem é economizar sem gerar previamente todas as respostas; a dependência é possuir supervisão representativa e recalibrar a seleção quando mudam modelos, tarefas ou critérios de qualidade.

O roteamento por similaridade compara a tarefa atual com exemplos anteriores e usa seu histórico para recomendar um modelo. Isso permite aproveitar a estrutura de domínios, mas proximidade no espaço de embeddings não equivale a igualdade de dificuldade ou de consequência. Uma questão simples e uma questão crítica podem compartilhar vocabulário e exigir tratamentos diferentes. A similaridade pode informar a expectativa de desempenho; a autorização para processar os dados deve vir de critérios próprios.

4.2 Decidir depois de observar uma resposta

Nas cascatas, um modelo é consultado antes que se decida encerrar ou escalar. O FrugalGPT explora combinações de modelos e critérios de aceitação para reduzir o custo de inferência [2]. A estratégia é atraente quando a primeira alternativa resolve parte expressiva das tarefas e o resultado pode ser avaliado de maneira suficientemente confiável. Quando quase tudo escala, o sistema paga pelo modelo inicial e pelo seguinte, além do avaliador e da latência acumulada.

O AutoMix combina uma resposta inicial, autoverificação e decisão sob incerteza, reconhecendo que a avaliação intermediária pode ser ruidosa [3]. A questão central deixa de ser apenas a capacidade do modelo pequeno e passa a incluir a capacidade do verificador de reconhecer seus erros. Um JSON válido pode conter dados errados, e uma resposta confiante pode carecer de sustentação. Validar formato, avaliar conteúdo e autorizar uma ação são verificações diferentes, ainda que compartilhem parte da infraestrutura.

Um LLM usado como juiz amplia a flexibilidade da avaliação, mas também introduz custo, dependência e possíveis erros correlacionados. Trocar o fornecedor do juiz pode ampliar diversidade, porém não demonstra independência efetiva: instruções, referências e critérios podem continuar compartilhados. Para tarefas importantes, o sistema precisa medir desacordos, falsos aceites e falsos bloqueios, além da nota média. A identidade do avaliador deve integrar o registro da decisão, sobretudo quando sua nota determina uma ação posterior.

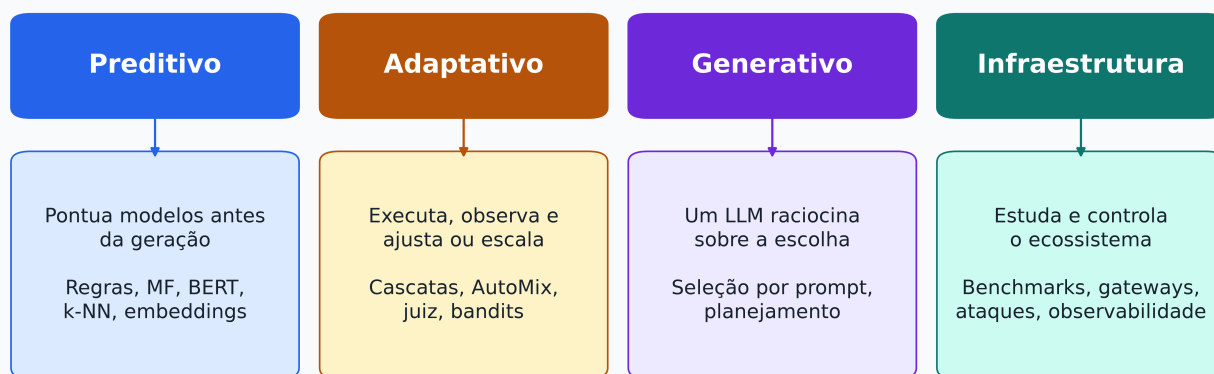
4.3 Adaptar a política e usar modelos para selecionar modelos

Estratégias adaptativas podem atualizar decisões a partir de resultados observados, enquanto estratégias generativas usam um LLM para interpretar, decompor ou planejar a própria seleção. Bandits e abordagens semelhantes enfrentam um compromisso adicional: explorar alternativas para aprender pode produzir custo e respostas inferiores durante a exploração. Um seletor generativo, por sua vez, acrescenta inferência antes da inferência principal e pode tornar a decisão mais difícil de explicar. Esses métodos precisam justificar o custo e o risco adicionais em relação a uma alternativa simples.

A revisão de Jiangeng e colaboradores organiza o campo em operações preditivas, adaptativas e generativas e distingue o trabalho de infraestrutura que avalia ou sustenta o roteamento [7]. Essa classificação complementa a divisão entre decidir antes e depois da geração. Um gateway pode combinar regras obrigatórias, classificador preditivo e escalonamento adaptativo sem que esses mecanismos sejam concorrentes em todos os aspectos. A taxonomia deve identificar qual decisão cada mecanismo toma e qual informação utiliza.

Figura 2. Taxonomia atualizada das estratégias de roteamento

Quatro paradigmas definidos pela operação algorítmica, conforme o survey sistemático de 2026.



Seleção → adaptação → raciocínio → avaliação e controle

Fonte: Jiangeng et al., Findings of EMNLP (2026)

© 2026 Aridio Silva | Project SGAEIA | CC BY 4.0

Figura 2. A taxonomia relaciona seleção prévia, cascatas e avaliação intermediária. Ela ajuda a identificar onde surgem chamadas adicionais e dependência de supervisão. As categorias podem coexistir numa aplicação; sua presença não determina uma ordem universal de qualidade.

Tabela 2 - painel 1/2

Estratégia	Chamadas de geração	Treinamento ou supervisão	Principal vantagem
Regras	Em geral, uma	Política e critérios explícitos	Transparência e baixo custo de seleção
Classificador ou preferência	Em geral, uma, além de eventuais embeddings	Dados rotulados e calibração	Seleção sem gerar respostas alternativas
Similaridade	Em geral, uma, além da representação vetorial	Histórico avaliado	Aproveitamento incremental de exemplos
Cascata	Uma ou mais	Critério de aceitação	Escalar somente quando necessário
Autoverificação ou juiz	Duas ou mais operações, conforme o desenho	Exemplos, rubrica ou avaliador	Avaliação condicionada ao resultado
Adaptativo ou generativo	Variável	Retorno observado ou instruções de seleção	Adaptação a estado e contexto

Tabela 2 - painel 2/2

Estratégia	Principal dependência
Regras	Cobertura das regras e tratamento de exceções
Classificador ou preferência	Representatividade dos dados
Similaridade	Cobertura e relevância dos vizinhos
Cascata	Confiabilidade e custo da validação
Autoverificação ou juiz	Erros correlacionados e custo adicional
Adaptativo ou generativo	Custo de exploração, estabilidade e explicação

5. Qualidade, custo e latência precisam da mesma unidade de análise

5.1 O custo relevante é o da tarefa concluída

O preço de uma chamada não descreve o custo de um sistema. A conta inclui tokens de entrada e saída, tentativas repetidas, embeddings, verificadores, infraestrutura e trabalho de operação. Em fluxos agênticos, também depende de quantas chamadas são necessárias para concluir uma tarefa e de quantas trajetórias precisam ser refeitas. Um modelo barato por token pode tornar a tarefa mais cara se exigir mais correções, gerar respostas excessivamente longas ou aumentar as chamadas a ferramentas.

Uma decomposição útil, proposta aqui para a avaliação operacional, é:

$$C_{\text{tarefa}} = C_{\text{selecao}} + C_{\text{geracao}} + C_{\text{validacao}} + C_{\text{repeticao}} + C_{\text{operacao}}$$

© 2026 Aridio Silva | Project SGAEIA | CC BY 4.0

Equação 1

Essa expressão organiza os componentes a medir; não presume que sejam constantes ou independentes. O custo de repetição depende da qualidade das respostas e da política de recuperação, enquanto o custo de operação pode conter uma parcela fixa relevante. Para comparar alternativas, é preciso manter a mesma definição de tarefa concluída e os mesmos critérios de aceitação. Sem isso, a economia pode resultar apenas de aceitar respostas piores ou encerrar tarefas prematuramente.

Considere um cenário hipotético em que a chamada ao modelo compacto custa US\$ 0,002 e a chamada ao modelo mais capaz custa US\$ 0,020, com cumprimentos de entrada e saída já incorporados nesses valores. Um seletor envia 80% das consultas ao primeiro e 20% ao segundo, acrescentando US\$ 0,0004 por consulta para decidir. O custo variável médio será US\$ 0,006, uma redução de 70% em relação ao envio de todas as consultas ao modelo mais caro. Esse cálculo pressupõe que a distribuição, os custos por chamada e a qualidade exigida permaneçam os mesmos.

$$C_{\text{roteado}} = 0,8(0,002) + 0,2(0,020) + 0,0004 = 0,006$$

© 2026 Aridio Silva | Project SGAEIA | CC BY 4.0

Equação 2

Se a operação do roteador acrescentar US\$ 500 mensais, a redução variável não se converte integralmente em economia total. Com 10 mil consultas, o gasto passa de US\$ 200 para US\$ 560; com 50 mil, de US\$ 1.000 para US\$ 800. O ponto de equilíbrio é aproximadamente 35.715 consultas mensais, sob as premissas do cenário. O exemplo demonstra por que volume e custo fixo devem acompanhar qualquer percentual de economia apresentado a quem decide a adoção.

Tabela 3

Volume mensal hipotético	Sempre o modelo mais caro	Roteamento, incluindo US\$ 500 fixos	Diferença para a referência
10.000 consultas	US\$ 200	US\$ 560	Aumento de US\$ 360
50.000 consultas	US\$ 1.000	US\$ 800	Economia de US\$ 200
100.000 consultas	US\$ 2.000	US\$ 1.100	Economia de US\$ 900

$$N^* = \frac{F}{C_{\text{base}} - C_{\text{roteado}}}$$

© 2026 Aridio Silva | Project SGAEIA | CC BY 4.0

Equação 3

O ponto de equilíbrio existe nessa forma somente quando o custo variável roteado é inferior ao custo da referência e a parcela fixa pode ser representada por F . Ele muda se o próprio gateway atender outros objetivos, como gestão de acesso, cujo custo já existiria sem seleção dinâmica. Nesse caso, deve-se avaliar o custo incremental de ativar o roteamento, evitando imputar duas vezes a mesma infraestrutura. A decisão econômica fica mais precisa quando separa o valor da mediação do valor do seletor.

5.2 Medir o compromisso sem confundir suas escalas

O RouteLLM define PGR como a fração da diferença de desempenho entre dois modelos recuperada pelo roteador. APGR resume a curva de recuperação ao variar o uso do modelo mais caro, e CPT indica a fração de chamadas necessária para alcançar uma recuperação especificada [1]. Esses indicadores ajudam a analisar uma configuração binária, mas não equivalem automaticamente a redução de cobrança em dólares. O custo real depende também dos comprimentos gerados, das tarifas e das operações auxiliares.

$$\text{PGR} = \frac{S_{\text{roteador}} - S_{\text{compacto}}}{S_{\text{forte}} - S_{\text{compacto}}}$$

© 2026 Aridio Silva | Project SGAEIA | CC BY 4.0

Equação 4

A expressão exige que a diferença no denominador seja relevante e diferente de zero. PGR de 50% significa recuperar metade da diferença entre as referências, não alcançar 50% da nota do modelo mais capaz. Se as notas das referências forem próximas, pequenas variações de avaliação podem alterar bastante o indicador. Por isso, a nota absoluta, a variabilidade e a curva de custo devem acompanhar o valor normalizado.

RouterBench e LLMRouterBench ampliam a análise para conjuntos de modelos e fronteiras de custo e qualidade [5][6]. Uma configuração é dominada quando outra oferece qualidade pelo menos igual com custo não superior, melhorando ao menos uma dimensão. A fronteira de Pareto reúne alternativas não dominadas, mas sua posição depende do conjunto comparado e das medidas escolhidas. Estar nessa fronteira não demonstra que uma alternativa respeite autorização, residência de dados ou requisitos de segurança.

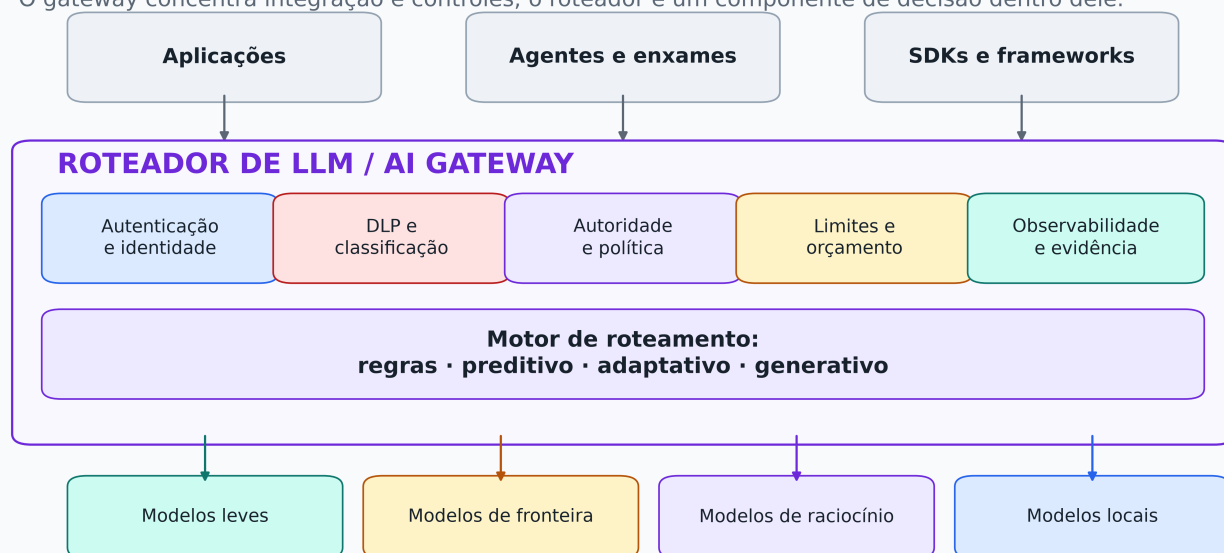
A latência precisa incluir seleção, fila, rede, geração e validação. Em cascatas, a sequência pode acumular esperas; na execução paralela, o tempo pode diminuir à custa de chamadas simultâneas. Devem ser observados o tempo até o primeiro token e o tempo total, com percentis como p50, p95 e p99 e sob uma carga declarada. O número de parâmetros do modelo, isoladamente, não determina o tempo percebido: infraestrutura, concorrência, contexto e extensão da saída podem alterar a comparação.

6. Integração e escolha de infraestrutura

Uma API comum reduz o trabalho de integração, mas compatibilidade sintática não significa equivalência funcional. Modelos e provedores podem diferir no tratamento de ferramentas, saídas estruturadas, imagens, contexto, parâmetros de amostragem e configurações de raciocínio. A aplicação precisa verificar as capacidades que utiliza e definir como tratar recursos ausentes. Uma troca bem-sucedida no transporte pode falhar na semântica da tarefa, mesmo quando a resposta HTTP indica sucesso.

Figura 3. Arquitetura de referência de um AI gateway

O gateway concentra integração e controles; o roteador é um componente de decisão dentro dele.



Fonte: LiteLLM/OpenRouter documentation; SGAEIA interpretation

© 2026 Aridio Silva | Project SGAEIA | CC BY 4.0

Figura 3. A arquitetura de referência organiza responsabilidades de mediação e observação em torno da seleção. Os componentes são papéis conceituais, que podem ser distribuídos em serviços e domínios de privilégio distintos. O gateway não substitui o controle de ferramentas nem a autorização das ações executadas pela aplicação.

O esforço de raciocínio pode integrar a configuração selecionada quando o provedor oferece esse controle. Nesse caso, a unidade avaliada deixa de ser somente o nome do modelo e passa a incluir modelo, versão, configuração e condições de execução. Uma mesma família pode produzir custos e tempos diferentes conforme o esforço, o contexto e a extensão da resposta. Os valores aceitos e seu significado precisam ser obtidos na documentação da versão utilizada, em vez de presumidos por uma interface multiprovedor.

O ecossistema também reúne ferramentas com finalidades diferentes. RouteLLM e LLMRouter oferecem recursos para desenvolver e avaliar seletores [1][8]; LiteLLM documenta mediação de acesso e gestão de chaves [10]; OpenRouter documenta substituição de modelos e seleção de provedores [12][13]. Portkey descreve funcionalidades de gateway, incluindo opções de resiliência e controle de tráfego [15]. A documentação comprova o recurso anunciado e suas condições declaradas, mas não oferece uma avaliação independente de adequação ao ambiente do leitor.

Tabela 4

Tipo de solução	Valor esperado	Responsabilidade que permanece com o adotante
Biblioteca de pesquisa	Implementações comparáveis e instrumentos de avaliação	Integração, autorização, operação e validação no domínio
Gateway operado pela organização	Controle sobre implantação, configuração e integração de políticas	Atualização, capacidade, isolamento, observabilidade e resposta a incidentes
Serviço gerenciado ou agregador	Redução do trabalho de integração e acesso a alternativas	Avaliação contratual, tratamento de dados, dependência e recuperação

Tipo de solução	Valor esperado	Responsabilidade que permanece com o adotante
Seleção incorporada a um produto	Adaptação já integrada à experiência de uso	Compreender limites de escolha, visibilidade e controles efetivamente disponíveis

Hospedagem própria não assegura soberania se embeddings, telemetria ou mecanismos de suporte transmitirem dados a terceiros. Um serviço gerenciado também não deve ser avaliado apenas pelo número de modelos disponíveis: importam localização efetiva, retenção, suboperadores, administração e possibilidade de exportar evidências. A lista de candidatos deve corresponder a implantações identificáveis, e não somente a nomes comerciais de modelos. Duas implantações do mesmo modelo podem ter condições de tratamento de dados diferentes.

O desempenho do gateway deve ser medido separadamente do tempo do modelo. Testes com servidor simulado ajudam a isolar sobrecarga de mediação, mas não descrevem toda a aplicação; autenticação, banco, logs, filtros e concorrência alteram o resultado. A documentação de benchmarks do LiteLLM deve ser lida com suas configurações e condições de carga [14]. Comparações publicadas por fornecedores concorrentes exigem ainda atenção a versões, paridade da configuração e interesse comercial, antes de se concluir que diferenças de milissegundos persistirão em produção.

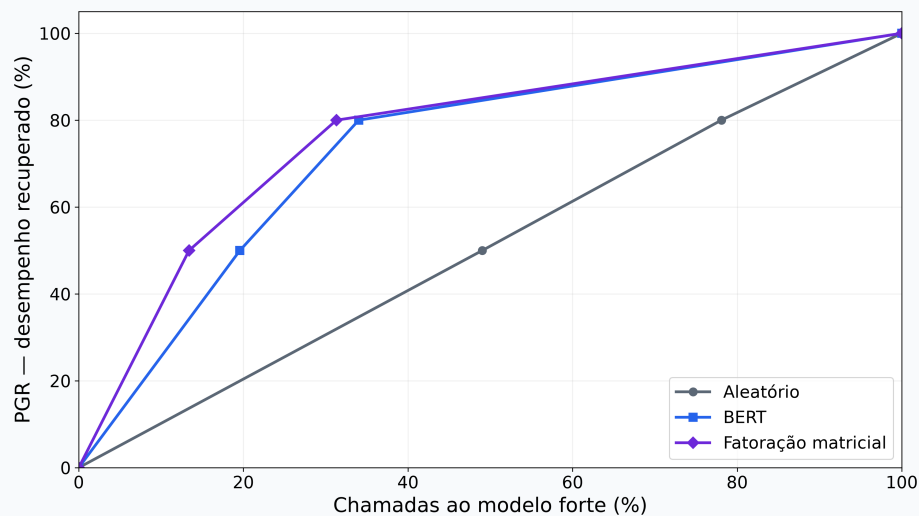
7. O que os benchmarks demonstram

Os estudos revisados sustentam a possibilidade de economizar com seleção de modelos, mas seus resultados respondem a perguntas diferentes. FrugalGPT relata reduções de custo de até 98% em configurações de cascata estudadas pelos autores [2]. Hybrid LLM relata até 40% menos chamadas ao modelo maior, e AutoMix relata redução superior a 50% no custo computacional para desempenho comparável nos cenários avaliados [3][4]. Esses números não formam uma classificação de superioridade: medem objetos diferentes e usam tarefas, modelos e avaliadores distintos.

O RouteLLM mostra o potencial de aprender a seleção com preferências. Na configuração de fatoração de matrizes com dados aumentados por juiz, a tabela de MT-Bench apresenta CPT de 13,40% para recuperar 50% da diferença de desempenho e de 31,31% para recuperar 80% [1]. A Figura 4 representa esse tipo de comparação entre fração de chamadas e recuperação da diferença. O resultado demonstra eficiência naquele protocolo; não determina a fração adequada para outro domínio ou outra geração de modelos.

Figura 4. Custo para recuperar desempenho no MT Bench

Pontos CPT(50%) e CPT(80%); segmentos são interpolação ilustrativa.



MF:

**13,4% das consultas para PGR de 50%;
31,3% das consultas para PGR de 80%.**

Fonte: Ong et al., RouteLLM, ICLR 2025, Table 1

© 2026 Aridio Silva | Project SGAEIA | CC BY 4.0

Figura 4. A comparação usa resultados de MT-Bench do estudo RouteLLM. CPT se refere à recuperação da diferença entre as duas referências, e não à porcentagem da nota absoluta do modelo forte. Os valores pertencem ao par de modelos, aos dados e à avaliação do estudo [1].

RouterBench contribui com resultados de inferência pré-computados e um protocolo para comparar políticas no espaço de custo e qualidade [5]. Esse desenho reduz o custo de testar seletores sobre as mesmas respostas e permite identificar ganhos que não dependem de uma nova geração a cada avaliação. Ao mesmo tempo, uma matriz estática não reproduz automaticamente filas, indisponibilidades, mudanças de API ou adaptações que ocorrem durante uma trajetória real. O benchmark torna uma comparação controlável; a avaliação operacional precisa acrescentar as dimensões ausentes.

LLMRouterBench oferece um contraponto importante aos estudos proponentes. Seu protocolo reúne 33 modelos e 21 conjuntos de dados, confirma complementaridade entre modelos e encontra vários métodos com desempenho semelhante, incluindo casos que não superam referências simples de forma confiável [6]. A conclusão útil é que a existência de alternativas complementares não assegura que um seletor consiga explorá-las. Melhorar o conjunto de modelos e a cobertura dos dados pode ser mais relevante que trocar o nome do algoritmo.

Tabela 5

Evidência	O que sustenta	O que não autoriza concluir
FrugalGPT, Hybrid LLM e AutoMix [2][3][4]	Existem configurações em que seleção ou escalonamento reduzem custo preservando a medida de qualidade adotada	Que os percentuais sejam transferíveis ou diretamente comparáveis
RouteLLM [1]	Preferências podem orientar seleção eficiente entre modelos	Que um limiar publicado sirva ao tráfego de outra organização
RouterBench [5]	Respostas pré-computadas permitem comparar políticas sob referências comuns	Que a simulação reproduza o comportamento completo do serviço
LLMRouterBench [6]	Complementaridade e dificuldade de superar referências simples podem coexistir	Que qualquer roteador aprendido supere uma política fixa
LLMRouter [8]	Uma infraestrutura comum permite estudar diferentes formulações de roteamento	Que disponibilidade de uma implementação demonstre prontidão operacional

A avaliação local deve incluir roteamento aleatório, sempre o modelo de menor custo, sempre o melhor modelo fixo para a amostra, regras simples e um classificador pequeno. Um oráculo retrospectivo pode

estimar o potencial do conjunto, usando a informação das respostas que não estaria disponível antes da execução. Esse oráculo é uma referência analítica e não um sistema implantável sem custo. A comparação precisa usar os mesmos dados, modelos e critérios de aceitação, mantendo dados de calibração separados dos utilizados para decidir se a estratégia trouxe ganho.

8. Comparação transversal das estratégias

As estratégias não devem ser escolhidas apenas pela sua melhor nota publicada. O mesmo método pode ser apropriado para uma tarefa repetitiva e inadequado para outra em que a verificação é difícil ou a consequência de um erro é elevada. Para tornar essa diferença explícita, o quadro abaixo compara necessidades operacionais, comportamento de validação e governabilidade. As classificações são uma síntese de engenharia baseada nos mecanismos discutidos, não resultados de um experimento comum realizado neste artigo.

Tabela 6 - painel 1/2

Estratégia	Qualidade e calibração	Custo e latência	Generalização
Regras	Crítérios explícitos; exceções precisam de revisão	Baixo custo de decisão; uma geração usual	Limitada pela cobertura da política
Classificador	Depende da supervisão e do limiar	Treinamento e representação somam custo; evita gerar alternativas	Precisa de avaliação fora da distribuição
Similaridade	Depende de exemplos comparáveis	Busca e embeddings podem ser relevantes	Sensível à cobertura da base
Cascata	Depende de reconhecer respostas inadequadas	Pode acumular geração, validação e escalonamento	Verificador também precisa generalizar
Juiz ou autoverificação	Exige rubrica e medida de falsos aceites	Chamadas adicionais podem dominar o custo	Pode repetir erros do gerador
Adaptativo ou generativo	Depende de retorno, estabilidade e contexto	Exploração ou planejamento acrescentam operações	Pode adaptar-se, com risco de instabilidade

Tabela 6 - painel 2/2

Estratégia	Auditabilidade
Regras	Boa explicação da regra aplicada
Classificador	Requer versão, sinais e decisão registrados
Similaridade	É possível registrar vizinhos e critérios
Cascata	Permite rastrear aceitação e escalada
Juiz ou autoverificação	Nota e justificativa não substituem evidência
Adaptativo ou generativo	Exige observar mudanças de política e estado

Tabela 7 - painel 1/2

Estratégia	Robustez adversarial	Agentes e enxames	Maturidade a verificar
Regras	Entradas podem explorar exceções e fronteiras	Úteis para restrições obrigatórias e limites	Cobertura e consistência das regras
Classificador	Sujeito a manipulação dos sinais de seleção [9]	Deve considerar papel, estado e consequência	Calibração no fluxo real
Similaridade	Vizinhança pode ser influenciada por conteúdo adversarial	Histórico e isolamento de contexto são decisivos	Cobertura e atualização da base

Estratégia	Robustez adversarial	Agentes e enxames	Maturidade a verificar
Cascata	Critério de escalada pode ser explorado	Precisa de orçamento para toda a trajetória	Taxa de escalonamento e qualidade do aceite
Juiz ou autoverificação	O avaliador também recebe conteúdo potencialmente hostil	Revisão cruzada não assegura independência	Erros do avaliador e correlação
Adaptativo ou generativo	A decisão de controle pode absorver instruções da tarefa	Maior contexto não elimina limites de autoridade	Estabilidade sob mudança e falha

Tabela 7 - painel 2/2

Estratégia	Complexidade operacional
Regras	Cresce com o número de exceções
Classificador	Dados, treino, versões e monitoramento
Similaridade	Gestão do índice e dos dados de supervisão
Cascata	Tratamento de estados e falhas parciais
Juiz ou autoverificação	Rubricas, reavaliação e desacordos
Adaptativo ou generativo	Exploração, reversão e observação contínua

O quadro mostra por que auditabilidade e qualidade são dimensões relacionadas, mas diferentes. Uma regra pode ser fácil de explicar e ainda selecionar mal; um classificador pode escolher bem numa distribuição e ser difícil de calibrar fora dela. Também mostra por que não existe uma ordem fixa de complexidade útil: o custo de manter centenas de exceções pode superar o de um classificador estável. A seleção deve considerar o ciclo completo de configuração, operação, investigação e atualização.

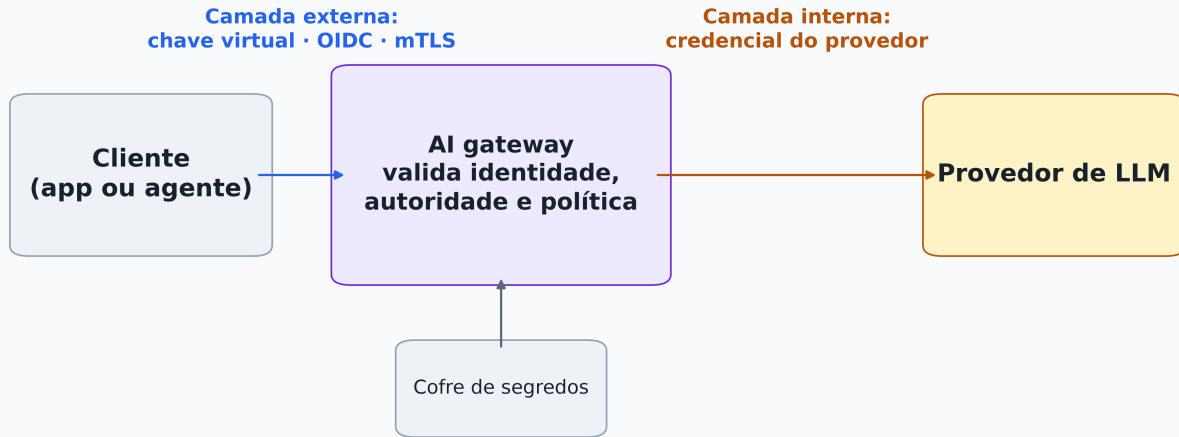
Para agentes, a questão decisiva é se a estratégia considera informações relevantes para a próxima ação. Para enxames, acrescentam-se concorrência, estado compartilhado e consequências acumuladas. A política de acesso e os limites coletivos devem continuar aplicáveis qualquer que seja o método de seleção. Essa separação permite melhorar o algoritmo sem transformar cada atualização num aumento implícito de autoridade.

9. Identidade, autoridade e substituição governada

Uma chave válida identifica uma credencial aceita pelo gateway. Ela não informa, por si só, se o titular pode enviar aqueles dados, usar determinada implantação ou desencadear uma ação. A distinção entre autenticação e autorização é central à arquitetura Zero Trust [16]. Em sistemas agênticos, também é necessário identificar em nome de quem o agente atua e qual parte da autoridade pode delegar.

Figura 5. Autenticação, identidade e autoridade em duas camadas

Credenciais do cliente e dos provedores são separadas; política não deve ser confundida com autenticação.



Fonte: Documentação de gateways; conceitos de Zero Trust do NIST

© 2026 Aridio Silva | Project SGAEIA | CC BY 4.0

Figura 5. A separação entre credenciais externas e internas reduz a necessidade de distribuir chaves de provedores aos clientes. Sua eficácia depende do isolamento do gateway, da autorização por objeto e da proteção das operações administrativas. Uma chave virtual continua sendo um segredo com alcance e ciclo de vida próprios.

Uma implantação deve limitar o conjunto de destinos acessíveis por usuário, serviço ou agente, além de proteger as credenciais que abrem acesso a esses destinos. Parâmetros fornecidos pelo cliente não devem conceder a capacidade de substituir endpoints administrativos ou redirecionar segredos para hosts arbitrários. A autorização também precisa abranger leitura de logs, alteração de orçamento e emissão de novas credenciais. A concentração de segredos no gateway reduz sua dispersão, mas aumenta o impacto potencial de uma falha no mediador.

O fallback merece atenção especial porque costuma ser apresentado como uma propriedade de disponibilidade. A documentação do OpenRouter descreve tentativas sucessivas de modelos conforme a prioridade e as condições de erro, incluindo situações de moderação [12]. Do ponto de vista da organização, a existência de uma alternativa tecnicamente acessível não a torna admissível para aquela requisição. Uma recusa de segurança tampouco deve ser tratada automaticamente como indisponibilidade a superar por um modelo menos restritivo.

O conjunto admissível pode ser representado por $A(x, h, i, p)$, em que x é a tarefa, h o histórico relevante, i a identidade e autoridade do solicitante, e p as políticas vigentes. Cada elemento desse conjunto deve representar uma implantação e configuração permitidas nas condições atuais. A otimização de qualidade, custo e latência ocorre dentro desse conjunto, e não por compensação de uma violação obrigatória com uma vantagem econômica. Trata-se de uma formalização de alto nível proposta para organizar a análise, sem especificar mecanismos privados do SGAEIA.

$$d^* = \arg \max_{d \in \mathcal{A}(x, h, i, p)} [\widehat{Q}(d) - \lambda_C \widehat{C}(d) - \lambda_L \widehat{L}(d)]$$

© 2026 Aridio Silva | Project SGAEIA | CC BY 4.0

Equação 5

Os pesos representam preferências entre alternativas já admissíveis; não atribuem preço a uma autorização ausente. Se o conjunto estiver vazio, a aplicação precisa de um comportamento definido, como suspender, reduzir a funcionalidade permitida ou encaminhar a decisão a uma instância autorizada. Se política, identidade ou estado mudarem, uma escolha antes válida pode deixar de ser válida. A revogação deve

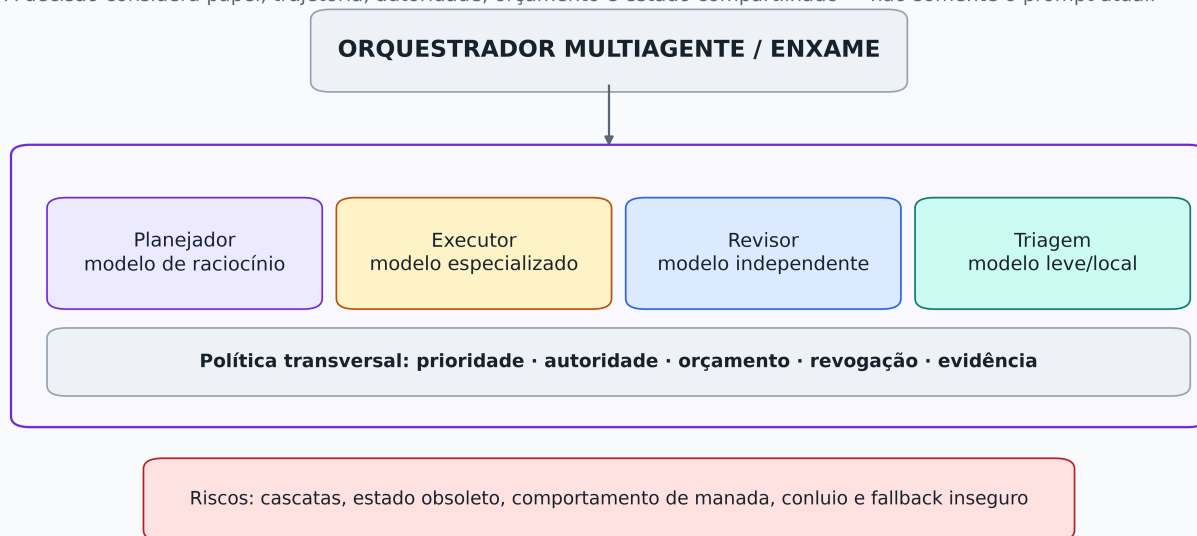
alcançar requisições em fila, caches e execuções em andamento conforme os limites definidos para a aplicação.

10. Agentes, multiagentes e enxames

Um agente pode executar várias interações com um modelo, usar ferramentas e revisar seu plano. Um sistema multiagente distribui parte desse trabalho entre agentes que trocam informações ou compartilham recursos; um enxame pode intensificar a descentralização e a coordenação coletiva. O número de agentes, sozinho, não determina criticidade. Uma única ação com amplo privilégio pode ser mais arriscada que muitas consultas limitadas, enquanto a combinação de ações aparentemente pequenas pode produzir um efeito que nenhuma delas revela isoladamente.

Figura 6. Roteamento governado em sistemas multiagentes e enxames

A decisão considera papel, trajetória, autoridade, orçamento e estado compartilhado — não somente o prompt atual.



Fonte: MTRouter (ACL 2026); OWASP Agentic Top 10 2026; interpretação SGAEIA

© 2026 Aridio Silva | Project SGAEIA | CC BY 4.0

Figura 6. A distribuição por papéis ilustra como diferentes agentes podem demandar capacidades distintas. Planejamento, execução e revisão não constituem, por si só, separação efetiva de funções. A análise também precisa considerar autoridade, dependências, memória, orçamento coletivo e possibilidade de conter a execução.

Selecionar modelos por papel é uma primeira aproximação: triagem pode exigir baixa latência; planejamento, maior capacidade de analisar alternativas; execução, aderência a formatos; revisão, capacidade de verificar o resultado. Essas associações precisam ser avaliadas por tarefa, evitando assumir que o modelo mais caro seja sempre o melhor revisor ou que um modelo pequeno baste para toda triagem. Uma classificação incorreta no início pode encaminhar dados ou autoridade de maneira inadequada antes de qualquer revisão. O roteamento sensível ao histórico amplia a informação disponível, mas não substitui controles de acesso [8].

Em trajetórias longas, medir apenas chamadas isoladas pode esconder gasto e risco acumulados. Se uma política reduz o custo por chamada em 30%, mas eleva em 60% o número de chamadas necessárias para concluir a tarefa, o custo variável torna-se 1,12 vez o anterior, sob essas hipóteses. A redução local transforma-se em aumento de 12% por tarefa. O exemplo aritmético mostra por que conclusão, retrabalho, ferramentas e validação precisam integrar a unidade de análise.

$$C_{\text{novo}}/C_{\text{base}} = 0,70 \times 1,60 = 1,12$$

© 2026 Aridio Silva | Project SGAEIA | CC BY 4.0

Equação 6

A concorrência acrescenta efeitos coletivos: várias tentativas de recuperação podem gerar uma onda de requisições, disputar a mesma cota e atrasar o caminho crítico. Limites por agente precisam ser combinados com limites por tarefa, equipe e serviço, pois contas individuais não impedem esgotamento agregado. Filas, contrapressão, cotas de concorrência e interrupção controlada devem considerar prioridades e justiça entre cargas. A resposta a uma falha precisa evitar que todos os agentes reproduzam simultaneamente a mesma decisão de escalonamento.

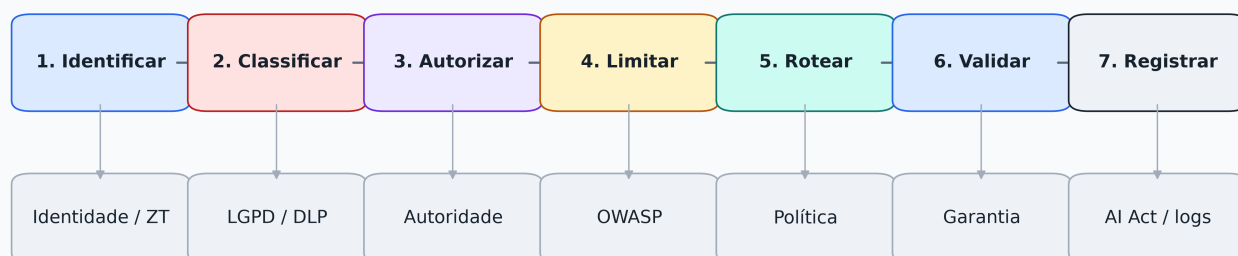
O cache exige cuidado semelhante. A documentação do LiteLLM alerta que o cache semântico voltado a prompts independentes pode reproduzir respostas obsoletas em tráfego multitermo ou agêntico [11]. Dois prompts parecidos podem corresponder a identidades, permissões, documentos ou estados diferentes. O benefício de reutilizar uma resposta só se sustenta quando o escopo da reutilização preserva essas condições, inclusive após revogação ou mudança de política.

A integridade da trajetória também exige observar relações entre ações. Um agente pode preparar dados, outro selecionar um destino e um terceiro transmitir o resultado; avaliar cada etapa isoladamente pode não revelar a transferência indevida produzida pelo conjunto. A taxonomia OWASP para aplicações agênticas reforça a necessidade de examinar riscos ligados a objetivos, ferramentas, identidade, memória e coordenação [24]. O vínculo com o SGAEIA está em preservar autoridade limitada e evidência ao longo dessas relações, em vez de considerar apenas a qualidade linguística de cada resposta [27].

11. Governança, segurança e evidências

Figura 7. Pipeline de governança e evidência no gateway

Os controles apoiam requisitos e auditoria; não constituem conformidade automática.



A evidência deve permanecer rastreável em toda a trajetória de decisão.

Fonte: NIST AI RMF; OWASP 2026; Regulation (EU) 2024/1689; LGPD; SGAEIA interpretation

© 2026 Aridio Silva | Project SGAEIA | CC BY 4.0

Figura 7. A relação entre classificação, política, execução e evidência mostra como o gateway pode apoiar a gestão de riscos. A correspondência com marcos de governança é interpretativa e depende da aplicação. Ela não demonstra conformidade jurídica nem certificação de um sistema de gestão.

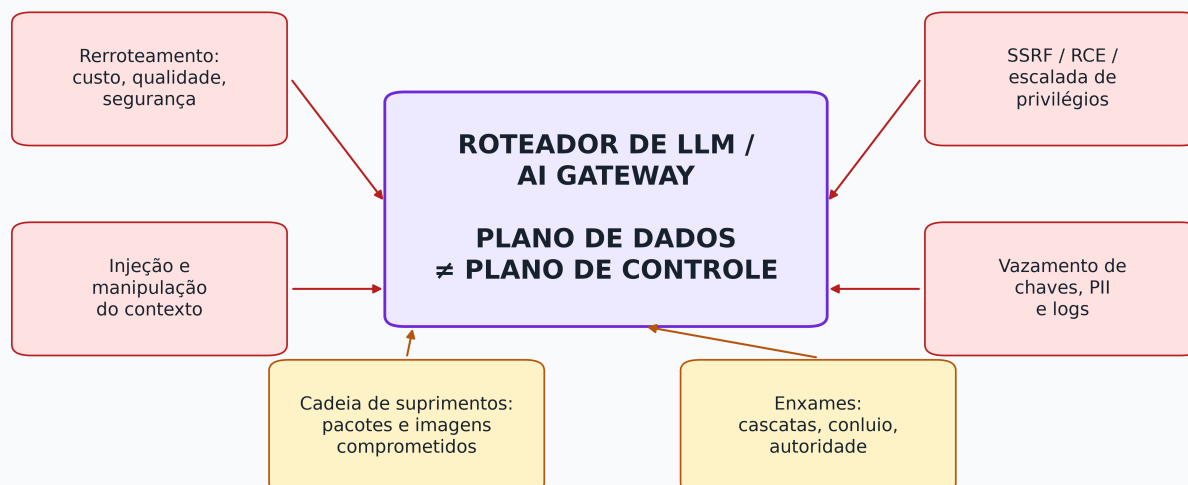
11.1 Um ponto de mediação também concentra risco

O gateway pode reduzir dispersão de credenciais e permitir controles consistentes, mas concentra tráfego, segredos e decisões valiosas para um atacante. O registro CVE-2026-42208 descreve injeção de SQL na verificação de chaves da API do proxy LiteLLM [25]. O relato sobre a versão 1.107.0 documenta um incidente de distribuição de pacote comprometido no ecossistema do projeto [26]. São evidências de classes de falha e condições específicas; não constituem uma avaliação comparativa da segurança de todos os gateways.

Há ainda ataques dirigidos ao seletor, que independem de uma vulnerabilidade tradicional de software. Shafran e colaboradores demonstram sequências que alteram o encaminhamento para modelos mais caros, inclusive em condições de caixa-preta, e discutem limites de filtragem por perplexidade [9]. Uma entrada pode preservar o sentido aparente da tarefa e mudar seu custo de execução. O monitoramento precisa observar decisões de roteamento e padrões de consumo, além do conteúdo da resposta final.

Figura 8. Modelo de ameaças do AI gateway e do roteador

A superfície inclui decisão, software, cadeia de suprimentos, credenciais e trajetórias multiagentes.



Fonte: OWASP 2026; Shafran et al. (2025); avisos do LiteLLM e incidentes de 2026

© 2026 Aridio Silva | Project SGAEIA | CC BY 4.0

Figura 8. O modelo organiza ameaças sobre entrada, decisão, credenciais, execução e observação. As relações indicam onde uma falha pode afetar outras responsabilidades. Os controles associados precisam ser verificados separadamente; uma caixa de proteção no diagrama não comprova prevenção de todos os ataques.

A taxonomia de consumo ilimitado do OWASP inclui cenários de esgotamento financeiro e degradação do serviço [23]. Orçamentos, limites de saída e de concorrência, detecção de desvios e tratamento de tentativas repetidas ajudam a limitar esse impacto. Esses controles também podem bloquear uso legítimo, especialmente durante picos ou tarefas excepcionalmente complexas. A avaliação deve registrar falsos bloqueios, recursos consumidos antes da contenção e a capacidade de recuperar o serviço sem reabrir o mesmo caminho de abuso.

11.2 Conformidade exige contexto e responsabilidade

NIST AI RMF organiza a gestão de riscos por funções de governar, mapear, medir e gerenciar; o perfil NIST AI 600-1 trata sua aplicação à AI generativa [17][18]. ISO/IEC 42001 trata do sistema de gestão de AI da organização [19]. Essas referências ajudam a distribuir responsabilidades e organizar evidências, mas não certificam automaticamente uma ferramenta ou um fluxo por possuir logs, filtros ou chaves virtuais. O enquadramento precisa considerar finalidade, pessoas afetadas, dependências e consequências.

A localização dos dados também não se resume à região declarada do modelo. Pode haver tratamento em embeddings, observabilidade, suporte e armazenamento de respostas, além da inferência. No contexto brasileiro, a LGPD e o regulamento de transferência internacional de dados da ANPD exigem examinar condições e mecanismos aplicáveis ao tratamento e à transferência [20][21]. Um filtro geográfico no gateway pode apoiar uma decisão de engenharia, mas não substitui a análise da operação e de suas bases jurídicas.

O artigo 12 do AI Act trata do registro de eventos para os sistemas de alto risco aos quais se aplica [22]. Essa referência ilustra a importância da rastreabilidade contextualizada, sem tornar toda aplicação de LLM um sistema de alto risco ou impor o mesmo desenho de log a todos os casos. O calendário e o enquadramento regulatório devem ser verificados na versão aplicável à operação. O papel do gateway é produzir parte da

evidência técnica necessária, sob responsabilidades que vão além do componente.

Um registro útil associa a tarefa à identidade, à política aplicável, à versão da configuração, ao destino efetivo e ao resultado da validação. O conteúdo necessário varia com o risco e deve respeitar minimização e retenção. Registrar indiscriminadamente prompts, respostas e segredos pode transformar observabilidade em uma superfície de exposição. O valor da trilha está em permitir reconstruir uma decisão relevante sem coletar mais dados do que sua finalidade exige.

Encadeamento de hashes pode revelar alterações em relação a uma referência confiável, mas não torna o armazenamento imutável. Um administrador comprometido pode reescrever registros e recalcular uma cadeia que não tenha proteção ou ancoragem independentes. Controle de acesso, retenção, proteção de chaves e mecanismos de verificação precisam ser avaliados em conjunto. A evidência também perde valor se omitir tentativas malsucedidas, mudanças de destino ou decisões tomadas durante uma indisponibilidade.

12. Quadro SGAEIA de análise de roteamento governado

12.1 A contribuição analítica e seu limite

O SGAEIA Governated AI Routing Assurance Model, desenvolvido neste artigo como quadro analítico, relaciona quatro objetos: a restrição que deve permanecer válida, a decisão de roteamento que pode afetá-la, a evidência necessária para examinar a decisão e a resposta a uma violação. Seu ponto de partida é a distinção entre capacidade, autoridade e evidência presente na perspectiva pública do SGAEIA [27]. A literatura de roteamento fornece mecanismos e resultados; Zero Trust e gestão de riscos fornecem antecedentes para a separação entre acesso, decisão e responsabilidade [16][17]. O quadro articula esses elementos para a avaliação de seleção e substituição em fluxos com modelos e agentes.

A contribuição não está em afirmar que roteadores possuem riscos, nem em reivindicar a invenção de autorização ou auditoria. Está em explicitar relações que uma avaliação exclusivamente econômica não representa e aplicá-las à mesma configuração que se deseja otimizar. A unidade de comparação passa a ser uma decisão admissível com consequência observável, incluindo a opção de não prosseguir quando os pressupostos não puderem ser preservados. Trata-se de síntese analítica proposta pelo autor para este artigo, sem atribuir ao quadro implementação normativa ou validação operacional do SGAEIA.

Tabela 8 - painel 1/2

Ameaça ou condição	Decisão afetada	Propriedade a preservar	Evidência exigida
Uso fora do escopo delegado	Seleção de modelo, dados e ferramentas	Autoridade limitada à tarefa	Identidade, delegação e decisão de autorização
Fallback para destino não permitido	Troca de implantação ou provedor	Continuidade das restrições	Destino inicial e efetivo, regra aplicável e motivo da troca
Exposição de dados	Escolha de inferência, embedding e telemetria	Minimização e tratamento autorizado	Classe do dado, destinos e condições de retenção
Escalada de custo adversarial	Modelo, esforço e repetição	Limites por tarefa e por organização	Consumo acumulado e decisões de escalada
Estado ou permissão revogados	Uso de cache, fila ou decisão anterior	Atualidade da autorização	Versão da política e tempos de revogação e execução
Falha do avaliador	Aceitar, escalar ou liberar uma ação	Critério de aceite adequado ao efeito	Rubrica, resultado e verificação posterior
Efeito coletivo não permitido	Distribuição de subtarefas	Limites sobre a trajetória compartilhada	Relações entre agentes, ações e recursos
Comprometimento do gateway	Seleção, acesso e registro	Separação de privilégios e recuperação	Versões, mudanças e registros protegidos

Ameaça ou condição	Decisão afetada	Propriedade a preservar	Evidência exigida
Perda ou adulteração de evidência	Investigar e liberar continuidade	Possibilidade de verificar decisões críticas	Integridade, origem e completude da trilha

Tabela 8 - painel 2/2

Ameaça ou condição	Métrica de avaliação	Resposta prevista
Uso fora do escopo delegado	Decisões aceitas fora do escopo	Bloquear ou interromper o alcance indevido
Fallback para destino não permitido	Substituições incompatíveis com a política	Usar alternativa admissível ou suspender
Exposição de dados	Transferências indevidas e dados excedentes	Impedir a saída e tratar a exposição
Escalada de custo adversarial	Gasto antes da contenção e falsos bloqueios	Conter a trajetória e rever a política
Estado ou permissão revogados	Ações posteriores à revogação e atraso de propagação	Invalidar reutilização e interromper ações afetadas
Falha do avaliador	Falsos aceites, falsos bloqueios e desacordos	Reavaliar e limitar o efeito autorizado
Efeito coletivo não permitido	Violações por tarefa e tempo até contenção	Conter o conjunto afetado e reconciliar estado
Comprometimento do gateway	Alcance do comprometimento e tempo de revogação	Isolar componentes e recuperar credenciais e políticas
Perda ou adulteração de evidência	Eventos críticos sem evidência suficiente	Reduzir a operação permitida e reconstruir o incidente

As métricas da matriz precisam de denominadores e critérios de observação. Uma taxa de fallback incompatível deve indicar quantas substituições foram examinadas e como se determinou a incompatibilidade; latência de revogação requer um marco de início e um evento observável de aplicação. Uma taxa igual a zero numa amostra não prova ausência de falhas em toda a operação. O quadro torna essas perguntas visíveis e evita usar o simples número de registros como medida de qualidade da evidência.

12.2 Aplicação a três decisões de engenharia

Considere uma tarefa com dados cuja política permite apenas uma implantação aprovada. Há um modelo equivalente numa região não permitida, com preço menor e disponibilidade superior. Uma comparação restrita a custo, qualidade e latência favoreceria a segunda alternativa; a análise de admissibilidade a exclui antes dessa otimização. Se a implantação aprovada falhar e não houver outra opção admissível, manter o serviço a qualquer custo alteraria a decisão de tratamento de dados, não apenas o provedor.

No segundo cenário, um agente inicia uma tarefa com orçamento compatível com poucas chamadas, mas passa a repetir correções e revisões entre vários modelos. Cada chamada respeita seu limite individual, e o fluxo ultrapassa o orçamento da tarefa. A evidência exigida pela matriz associa consumo e resultado à trajetória, permitindo reconhecer a falha que o indicador por chamada não revela. A resposta apropriada pode ser interromper tentativas sem progresso, reduzir a funcionalidade permitida ou obter nova autorização para ampliar o trabalho.

No terceiro cenário, dois agentes usam respostas semanticamente semelhantes para decidir ações sobre documentos diferentes. O cache devolve uma resposta produzida com outro estado de permissões, e a similaridade textual parece justificar a reutilização. A análise distingue eficiência de cache de validade da decisão: identidade, estado e escopo precisam continuar compatíveis. Uma resposta anterior correta pode ser inadequada para o novo contexto sem que seu texto tenha mudado.

Tabela 9

Cenário	Conclusão por um indicador isolado	Conclusão com o quadro proposto	Evidência decisiva
Substituição de implantação	Menor custo e maior disponibilidade justificam a troca	A alternativa precisa satisfazer as restrições da tarefa	Política aplicável e destino efetivamente usado
Repetição multiagente	Todas as chamadas estão dentro do limite	O gasto agregado e a conclusão da tarefa precisam ser observados	Consumo por trajetória, tentativas e progresso
Reutilização de resposta	Alta similaridade favorece o cache	A validade depende também do contexto e da autorização	Estado, identidade e escopo da resposta reutilizada

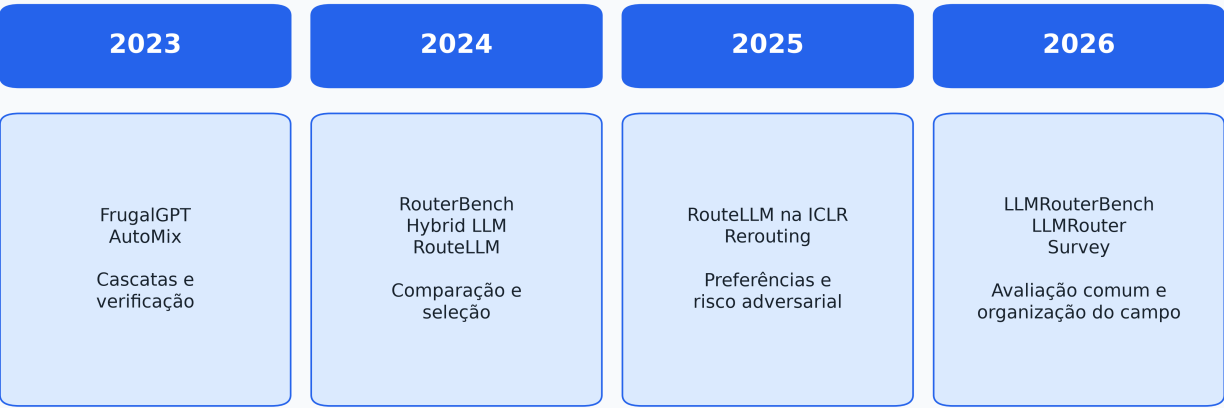
Esses casos demonstram a utilidade analítica do quadro: ele identifica a informação ausente e explica por que uma decisão economicamente atraente pode ser inadequada. Não medem frequência de incidentes nem quantificam redução de risco em produção. A verificação empírica deverá observar se os controles identificam violações e preservam tarefas legítimas, com seus custos e erros. Para o artigo técnico, a contribuição fica delimitada na comparação, na rastreabilidade das decisões e nos contraexemplos apresentados.

13. Evolução e maturidade devem ser avaliadas por dimensão

O desenvolvimento do campo combina linhas que amadurecem em ritmos diferentes. Cascatas e seleção por custo ganharam formulações explícitas em trabalhos como FrugalGPT, AutoMix e Hybrid LLM; RouteLLM ampliou o estudo da seleção com preferências; RouterBench e LLMRouterBench fortaleceram comparações comuns [1][2][3][4][5][6]. Em paralelo, a análise adversarial mostrou que a camada de seleção também pode ser manipulada [9]. Esses movimentos não descrevem uma progressão automática para segurança: cada capacidade exige sua própria evidência.

Figura 9. Marcos selecionados da pesquisa de roteamento

Métodos, avaliação e integridade da seleção; periodização interpretativa do autor.



Fonte: FrugalGPT; AutoMix; RouterBench; Hybrid LLM; RouteLLM; LLMRouterBench; Rerouting

© 2026 Aridio Silva | Project SGAEIA | CC BY 4.0

Figura 9. A periodização destaca trabalhos sobre economia, comparação e integridade da seleção. Os marcos indicam publicações e temas selecionados, sem representar uma história exaustiva ou medir adoção de mercado. A separação por fases é uma organização interpretativa do autor.

Figura 10. Pesquisa e governança por ano

As datas identificam publicações selecionadas; não medem adoção nem prontidão.

	2023	2024	2025	2026
Métodos	FrugalGPT AutoMix	Hybrid LLM RouteLLM	RouteLLM ICLR	LLMRouter
Avaliação		RouterBench		LLMRouterBench Survey
Segurança			Rerouting	
Governança	NIST AI RMF	NIST AI 600-1	OWASP Agentic 2026 (dez. 2025)	

Fonte: Estudos originais; NIST AI RMF; NIST AI 600-1; OWASP Agentic 2026

© 2026 Aridio Silva | Project SGAEIA | CC BY 4.0

Figura 10. As faixas mostram que métodos, avaliação, segurança e governança tratam dimensões complementares. As divisões anuais facilitam a leitura cronológica; proximidade temporal não implica que uma publicação tenha causado outra. Datas se referem aos marcos indicados, e não à prontidão de produtos.

A disponibilidade de um recurso documentado é evidência diferente de estabilidade sob carga, avaliação independente ou resistência a ataques. Uma organização pode dominar API, quotas e fallback e ainda não possuir dados para calibrar um seletor aprendido. Pode também obter boa qualidade média e desconhecer o comportamento após revogação, atualização de modelo ou perda de telemetria. A maturidade deve ser registrada por dimensão e pelo ambiente observado, evitando converter uma nota editorial em certificação externa.

Figura 11. Dimensões de prontidão e evidências necessárias

Disponibilidade funcional, medição local e condições adversas devem ser distinguidas.

Dimensão	Disponibilidade	Medição local	Condições adversas
Acesso e identidade	Recursos e escopos	Isolamento observado	Revogação e abuso
Seleção e qualidade	Modelos e política	Qualidade por tarefa	Mudança de domínio
Custo e latência	Limites e orçamento	Custo total e caudas	Picos e repetição
Substituição	Alternativas definidas	Destino efetivo	Falha parcial
Agentes e exames	Papéis e coordenação	Trajatória completa	Efeito coletivo
Evidência	Eventos registrados	Compleitude e origem	Perda e adulteração

Fonte: Quadro analítico do autor

© 2026 Aridio Silva | Project SGAEIA | CC BY 4.0

Figura 11. O quadro distingue disponibilidade funcional, medição local e verificação sob condições adversas. Ele orienta perguntas de prontidão, sem atribuir um nível universal a um produto ou ao mercado. As linhas e colunas permitem examinar lacunas que uma classificação única ocultaria.

O custo da manutenção merece o mesmo cuidado. Mudanças em modelos, preços, parâmetros e padrões de tráfego podem invalidar limiares e tornar o histórico menos representativo. A gestão de versões deve associar uma decisão à configuração realmente executada, inclusive quando um alias aponta para um modelo atualizado. O processo de atualização precisa permitir comparar, limitar a exposição e recuperar uma configuração anterior quando a nova alternativa não atender aos critérios definidos.

14. Quando adotar e quando manter uma solução mais simples

Um gateway pode ser justificado por necessidades de acesso, orçamento e observação mesmo que o tráfego permaneça num modelo fixo. Um seletor dinâmico exige uma justificativa adicional: o ganho obtido com tarefas heterogêneas deve superar seu custo e sua complexidade. Se um único modelo atende bem ao domínio, o volume é baixo ou os dados para calibração são insuficientes, uma política fixa pode ser a escolha mais consistente. A adoção deve responder a um problema observado, como gasto, latência ou capacidade de governar o acesso.

O primeiro passo é definir conclusão e qualidade aceitável por classe de tarefa. O segundo é determinar destinos, dados e configurações admissíveis, antes de otimizar suas combinações. Em seguida, deve-se medir referências simples e avaliar uma estratégia candidata nas mesmas condições, incluindo o custo da decisão e da validação. A comparação precisa preservar tarefas difíceis e classes pouco frequentes, pois uma média favorável pode esconder a deterioração de um subconjunto importante.

Tabela 10

Condição observada	Decisão inicial razoável	Evidência para avançar
Necessidade de centralizar acesso e gastos	Avaliar mediação com modelo fixo	Isolamento, limites e custo operacional
Tarefas repetitivas e baixo volume	Manter modelo ou regras simples	Qualidade suficiente e economia total demonstrada

Condição observada	Decisão inicial razoável	Evidência para avançar
Tráfego heterogêneo e volume relevante	Comparar seletor preditivo com referências	Curva de qualidade e custo por tarefa
Resultado verificável com confiabilidade	Avaliar cascata	Falsos aceites, escalonamento e latência total
Dados ou ações com restrições relevantes	Restringir candidatos antes da seleção	Continuidade de política e evidência de substituição
Agentes com muitas etapas	Avaliar a trajetória completa	Conclusão, custo agregado, autoridade e contenção
Pouca visibilidade sobre falhas e mudanças	Melhorar observação antes de automatizar novas decisões	Registro do destino efetivo e recuperação verificável

A entrada em operação deve usar exposição gradual e critérios prévios de continuidade. Custo por tarefa, aceitação de respostas, retrabalho, latência de cauda, violações de política e completude da evidência precisam ser observados em conjunto. Quando o comportamento muda, pode ser necessário recalibrar, restringir o conjunto de modelos ou retornar a uma política mais simples. A possibilidade de recuar faz parte do projeto da adoção, assim como a possibilidade de escalar.

15. Discussão e limites da análise

As seis perguntas iniciais permitem reunir os resultados sem reduzir o artigo a um catálogo. Há condições favoráveis para economia, mas a complementaridade dos modelos é apenas uma oportunidade; o seletor precisa conseguir explorá-la. O gateway oferece valor operacional próprio, ao mesmo tempo que concentra responsabilidades e amplia o alcance de uma falha. Em agentes, o resultado relevante é uma tarefa concluída sob seus limites, e não uma coleção de respostas individualmente bem avaliadas.

Tabela 11

Pergunta orientadora	Resposta sustentada pela análise
Quando economiza preservando qualidade?	Quando heterogeneidade, calibração e capacidade de validação compensam decisão, repetição e operação
Quando supera referências simples?	Quando o ganho aparece no mesmo protocolo e permanece nas classes de tarefa relevantes; complexidade não o assegura
Que valor independe do algoritmo?	Mediação de acesso, gestão de recursos, observação e aplicação consistente de controles
Que risco decorre da concentração?	Uma falha pode alcançar credenciais, dados, configuração e evidências de muitas aplicações
O que muda com agentes e enxames?	Estado, delegação, concorrência e efeitos acumulados tornam a trajetória uma unidade necessária de avaliação
Que evidência permite governar?	Registros verificáveis de autoridade, política, destino efetivo, consumo, validação e resposta a mudanças

Os resultados publicados continuam limitados pelos modelos, tarefas e avaliadores de seus protocolos. Uma matriz de respostas pode favorecer análise econômica sem capturar comportamento de filas, ferramentas e falhas parciais; uma nota de juiz pode medir preferência sem representar correção ou segurança. Documentação de produto descreve condições declaradas pelo fornecedor, e relatos de incidentes não estimam a prevalência de vulnerabilidades no mercado. A análise deve preservar essas diferenças para que o leitor não transforme evidência localizada em promessa geral.

O quadro proposto também possui limites. Ele depende de políticas identificáveis, observação suficiente e uma definição do que constitui violação ou conclusão válida. Não elimina erros de classificação de risco, lacunas nas políticas ou comprometimento de componentes que deveriam aplicar os controles. Seu uso permite estruturar a avaliação e tornar explícita a informação ausente; demonstrar redução de risco exige

observar controles funcionando e falhando em condições previamente definidas.

Há questões abertas que merecem avaliação futura: seleção conjunta de modelo e configuração; efeitos da revogação sobre trajetórias já iniciadas; robustez do avaliador; custo de evidência e minimização; comportamento sob falhas simultâneas; e transferência para novos modelos e domínios. Essas questões derivam das limitações identificadas, em vez de serem apresentadas como resultados alcançados. A contribuição do artigo fica preservada ao distinguir a orientação prática disponível das hipóteses que ainda precisam de experimentação.

16. Conclusão

Roteadores de LLMs tornam a heterogeneidade de modelos uma possibilidade de decisão por tarefa, mas não asseguram economia nem qualidade por sua mera presença. Estratégias preditivas, cascatas e seleção adaptativa utilizam informações diferentes e distribuem custo e risco de maneiras distintas. A evidência publicada sustenta ganhos em configurações específicas e, simultaneamente, a dificuldade de superar referências simples em avaliações comuns. A consequência é exigir comparação local com critérios coerentes de conclusão, qualidade e custo total.

A análise também estabelece que a decisão deve ocorrer sobre alternativas admissíveis. Fallback, troca de modelo, reutilização de resposta e ampliação de esforço podem alterar tratamento de dados, consumo e continuidade da autorização. Em sistemas multiagentes, essas mudanças precisam ser acompanhadas na trajetória e nos recursos compartilhados. Um componente que melhora disponibilidade enquanto viola as condições da tarefa não preserva o resultado que a organização pretendia obter.

O quadro SGAEIA de análise de roteamento governado explicita essa relação entre restrição, decisão, evidência e resposta. Sua aplicação aos três cenários mostra informações e consequências que custo por chamada, similaridade textual ou disponibilidade isolados não revelam. Essa é a contribuição analítica do artigo: reunir a comparação técnica e a governança no mesmo processo de avaliação, com limites que podem ser examinados. Adotar roteamento significa, portanto, demonstrar utilidade sob condições definidas e manter a capacidade de observar, restringir, revogar e revisar suas decisões.

Apêndice A — Glossário técnico

Admissibilidade — Condição de uma implantação ou configuração satisfazer as restrições aplicáveis à tarefa antes da otimização.

Agente de AI — Componente que utiliza capacidade de decisão ou geração para conduzir etapas de uma tarefa, eventualmente com ferramentas e estado persistente.

AI gateway — Camada de mediação de acesso a modelos e serviços, que pode reunir adaptação, controles e observabilidade.

APGR — Resumo da recuperação de desempenho ao longo da variação do uso do modelo mais caro no protocolo RouteLLM [1].

Autenticação — Verificação de uma identidade ou credencial apresentada.

Autorização — Decisão sobre o acesso ou ação permitidos a uma identidade nas condições aplicáveis.

Baseline — Referência de comparação, como política fixa, regras simples ou seleção aleatória.

Cache de prefixo — Reutilização de processamento de partes comuns de entrada quando suportada pelo serviço.

Cache semântico — Reutilização de resposta com base em similaridade de entradas; exige examinar a validade do contexto reutilizado.

Cascata — Execução que consulta uma alternativa e decide, a partir do resultado ou da avaliação, se deve escalar.

Chave virtual — Credencial de mediação associada a permissões e limites, distinta da credencial do provedor.

Contrapressão — Controle que reduz ou adia a entrada de trabalho quando a capacidade disponível é insuficiente.

CPT — Fração de chamadas ao modelo forte necessária para uma meta de recuperação da diferença de desempenho [1].

Delegação — Concessão limitada de autoridade para executar parte de uma tarefa em nome de outra identidade.

Denial of Wallet — Consumo abusivo que explora a cobrança por uso para esgotar recursos financeiros [23].

Embedding — Representação vetorial usada para comparar ou caracterizar conteúdo; similaridade não demonstra equivalência de autorização.

Esforço de raciocínio — Configuração de recursos de inferência oferecida por determinados modelos, cujo significado depende da interface e da versão.

Fallback — Tentativa de uma alternativa quando a execução inicial não é concluída nas condições definidas; requer preservar as restrições da tarefa.

Fronteira de Pareto — Conjunto de alternativas não dominadas nos critérios considerados.

Implantação — Instância de acesso a um modelo, associada a provedor, região, versão e condições operacionais.

LLM-as-a-Judge — Uso de um modelo de linguagem para avaliar uma resposta ou apoiar uma decisão de aceitação.

Oráculo — Referência retrospectiva que utiliza informação indisponível ao seletor antes de executar ou observar as respostas.

PGR — Fração da diferença de desempenho entre duas referências recuperada pelo roteador [1].

Plano de controle — Responsabilidades de administrar políticas, configuração, permissões e ciclo de vida.

Plano de dados — Caminho que executa as requisições e transporta os dados conforme as decisões aplicáveis.

Revogação — Retirada de uma permissão ou condição de acesso, com necessidade de propagação às execuções afetadas.

Roteador preditivo — Seletor que decide antes de gerar a resposta principal.

Roteamento adaptativo — Seleção que ajusta decisões com base em resultados ou informação observada durante a execução.

Roteamento generativo — Uso de um modelo generativo para apoiar ou produzir a decisão de seleção.

Trajetória — Sequência relacionada de decisões, interações, ferramentas e mudanças de estado necessárias a uma tarefa.

TTFT — Tempo decorrido até o primeiro token, distinto do tempo necessário para concluir a resposta ou tarefa.

Zero Trust — Perspectiva de arquitetura que evita confiança implícita por localização ou propriedade do recurso e distingue autenticação de autorização [16].

Referências

1. ONG, Isaac et al. RouteLLM: Learning to Route LLMs with Preference Data. ICLR, 2025. arXiv:2406.18665v4. Texto integral.
2. CHEN, Lingjiao; ZAHARIA, Matej; ZOU, James. FrugalGPT: How to Use Large Language Models While Reducing Cost and Improving Performance. Preprint, 2023. arXiv:2305.05176.
3. AGGARWAL, Pranjal et al. AutoMix: Automatically Mixing Language Models. NeurIPS, 2024; versão arXiv v5, 2025. arXiv:2310.12963.
4. DING, Dujian et al. Hybrid LLM: Cost-Efficient and Quality-Aware Query Routing. ICLR, 2024. arXiv:2404.14618.
5. HU, Qitian Jason et al. RouterBench: A Benchmark for Multi-LLM Routing System. Preprint, 2024. arXiv:2403.12031.
6. LI, Hao et al. LLMRouterBench: A Massive Benchmark and Unified Framework for LLM Routing. Findings of ACL, 2026, p. 37733–37754. Registro e texto.
7. JIANGENG, Chang; WU, Xingyu; TAN, KC; FENG, Mengling. A Systematic Survey of Large Language Model Routing. Findings of EMNLP, 2026, p. 44102–44134. Registro e texto.
8. FENG, Tao et al. LLMRouter: Unified Infrastructure for Developing, Evaluating, and Deploying LLM Routers. Preprint, 2026. arXiv:2608.06867.
9. SHAFRAN, Avital; SCHUSTER, Roei; RISTENPART, Thomas; SHMATIKOV, Vitaly. Rerouting LLM Routers. Preprint, 2025. arXiv:2501.01818.
10. BERRIAI. Virtual Keys. Documentação do LiteLLM. Documento oficial. Consulta: 29 out. 2026.
11. BERRIAI. Semantic Caching. Documentação do LiteLLM. Documento oficial. Consulta: 29 out. 2026.
12. OPENROUTER. Model Fallbacks. Documentação oficial. Consulta: 29 out. 2026.
13. OPENROUTER. Provider Routing. Documentação oficial. Consulta: 29 out. 2026.
14. BERRIAI. Benchmarks. Documentação do LiteLLM. Documento oficial. Consulta: 29 out. 2026.
15. PORTKEY. AI Gateway. Documentação oficial. Consulta: 29 out. 2026.
16. ROSE, Scott; BORCHERT, Oliver; MITCHELL, Stu; CONNELLY, Sean. Zero Trust Architecture. NIST SP 800-207, 2020. DOI: 10.6028/NIST.SP.800-207.
17. NIST. Artificial Intelligence Risk Management Framework — AI RMF 1.0. NIST AI 100-1, 2023. Fonte oficial.
18. NIST. Artificial Intelligence Risk Management Framework: Generative Artificial Intelligence Profile. NIST AI 600-1, 2024. DOI: 10.6028/NIST.AI.600-1.
19. ISO; IEC. ISO/IEC 42001:2023 — Information technology — Artificial intelligence — Management system. Registro oficial.
20. BRASIL. Lei nº 13.709/2018 — Lei Geral de Proteção de Dados Pessoais, art. 33. Legislação federal.
21. ANPD. Resolução CD/ANPD nº 19, de 23 de agosto de 2024 — Regulamento de Transferência Internacional de Dados. Fonte oficial.
22. UNIÃO EUROPEIA. Regulamento (UE) 2024/1689 — AI Act, art. 12, versão consolidada disponibilizada pela Comissão Europeia. AI Act Service Desk.
23. OWASP GENAI SECURITY PROJECT. LLM10:2025 — Unbounded Consumption. Fonte oficial.
24. OWASP GENAI SECURITY PROJECT. OWASP Top 10 for Agentic Applications for 2026. 9 dez. 2025. Fonte oficial.
25. GITHUB ADVISORY DATABASE. LiteLLM has SQL Injection in Proxy API key verification — CVE-2026-42208 / GHSA-r75f-5x8p-qvmc. Registro. Consulta: 29 out. 2026.
26. BERRIAI/LITELLM. Security report concerning compromised litellm 1.107.0 distribution. Issue nº 47826. Relato técnico e discussão no repositório do projeto. Registro original. Consulta: 29 out. 2026.

Sobre o Autor

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

Sua pesquisa concentra-se em Agentic AI, Sistemas Multiagentes, Edge AI, Segurança de AI, Zero Trust, Security-by-Design, Governança de AI, Spec-Driven Development e garantia contínua de segurança.

É criador e pesquisador principal do SGAEIA — Secure Governed Autonomous Edge Intelligence Architecture, iniciativa aberta de pesquisa sobre fundamentos arquiteturais para sistemas de AI autônomos seguros, governados, auditáveis e confiáveis em ambientes distribuídos edge-cloud.

Perfis de Pesquisa

Aridio Silva — Pesquisador Independente, Brasil

- ORCID: <https://orcid.org/0009-0008-2411-6995>
- Google Scholar: <https://scholar.google.com/citations?user=rPn5O48AAAAJ>
- Zenodo — Comunidade SGAEIA: <https://zenodo.org/communities/sgaeia>
- OpenAIRE: <https://explore.openaire.eu/search/find?fv0=Aridio%20Silva&f0=q>
- Medium: <https://medium.com/@aridiosilva>
- GitHub: <https://github.com/aridiosilva>
- LinkedIn: <https://www.linkedin.com/in/aridio-silva-74997111/>
- Repositório público SGAEIA: <https://github.com/aridiosilva/SGAEIA>
- Artefato de Pesquisa SGAEIA / DOI: <https://doi.org/10.5281/zenodo.22557796>
- Homepage: <https://aridiosilva.com>
- Homepage SGAEIA: <https://aridiosilva.com/sgaeia>
- SGAEIA no LinkedIn: <https://www.linkedin.com/company/sgaeia/>
- Kaggle: <https://www.kaggle.com/aridiosilva2>
- DEV Community: <https://dev.to/aridiosilva>
- Academia.edu: <https://independent.academia.edu/AridioSilva>

Figuras

A capa não é numerada. As Figuras 1–11 são numeradas sequencialmente, citadas no texto e fornecidas em PNG e SVG. Os ativos conceituais permanecem dentro da fronteira de divulgação pública e são licenciados sob CC BY 4.0. A composição visual da capa foi feita com assistência de AI.

Licença

Exceto onde indicado de outra forma, o texto e as ilustrações conceituais originais deste artigo são licenciados sob a Creative Commons Attribution 4.0 International License (CC BY 4.0).

© Aridio Silva @aridiosilva · Outubro de 2026. O software e o artefato de pesquisa SGAEIA permanecem sujeitos à sua própria licença Apache License 2.0.

Autonomous AI. Governed by Design. Trusted by Evidence.

Continuidade da Série

Este trabalho é o Artigo 24 da Série de Pesquisa SGAEIA. O catálogo público da série está disponível em <https://aridiosilva.com>, com a presença do projeto em <https://aridiosilva.com/sgaeia>.

Registro persistente do Artigo 24: <https://doi.org/10.5281/zenodo.23763978>. 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. Roteadores de LLMs para Sistemas de AI: Arquitetura, Governança, Multiagentes, Enxames e Segurança — artigo atual.

Citação Sugerida

Silva, Aridio. (2026). Roteadores de LLMs para Sistemas de AI: Arquitetura, Governança, Multiagentes, Enxames e Segurança. Série de Pesquisa SGAEIA, Artigo 24. DOI: 10.5281/zenodo.23763978.