

Do LLM ao sistema agêntico

Como um modelo de linguagem colabora com um coordenador, um RAG documental e ferramentas para realizar uma tarefa.

Inteligência Artificial • Prof. Francisco Javier Ropero Peláez • UFABC

Ideia central: o modelo interpreta e propõe; o coordenador verifica e encaminha; a ferramenta executa; o resultado volta ao modelo. A pessoa define os objetivos, os limites e a revisão.

Comece pelo conjunto, não por uma coleção de siglas

Imagine uma pessoa que já usa um modelo de linguagem e quer investigar uma questão. O modelo pode organizar ideias e sugerir um plano, mas não recebe automaticamente seus documentos, não dispõe de todos os instrumentos de cálculo e não adquire permissão para alterar arquivos só por escrever uma instrução. O aplicativo acrescenta essas capacidades por meio de componentes de software.

O LLM é uma peça do sistema, frequentemente baseada na arquitetura Transformer. A passagem para um sistema agêntico não exige transformar os pesos da rede em um novo tipo de programa. O que muda é a organização externa: suas propostas podem selecionar operações permitidas, e as observações dessas operações alimentam decisões seguintes. [T1, T2]

Pessoa: objetivo e limites

↓

Coordenador ↔ LLM

↓

Ferramentas • recuperação • dados • estado

↓

Observações → coordenador → LLM → próxima ação ou resposta

O desenho mostra responsabilidades, não a localização física dos componentes. Modelo e ferramentas podem funcionar em computadores diferentes. Nenhum dos módulos precisa estar dentro de uma cabeça de atenção.

O que você aprenderá

Distinguir texto de ação; acompanhar uma ida e uma volta completas; interpretar JSON; compreender RAG e recuperação documental; separar memória de projeto de parâmetros do modelo; e explicar quem controla a execução.

1 • ARQUITETURA GERAL

Cada componente resolve uma necessidade

Começamos com o LLM e acrescentamos componentes somente quando uma necessidade aparece. Esta é uma organização possível, não uma lista obrigatória para toda aplicação. [T1]

Necessidade	Componente e responsabilidade
Receber o objetivo e mostrar o andamento	Interface: formulário, chat, botões e apresentação do resultado.
Informar o que está disponível agora	Preparação de contexto: reúne instruções, objetivo, contratos e observações pertinentes.
Interpretar o objetivo e propor o próximo passo	LLM: produz uma mensagem; pode selecionar uma ação dentre as oferecidas.
Controlar a execução	Coordenador: interpreta a proposta, verifica formato, argumentos e permissões, chama a função e registra sua saída.
Obter informações fora da mensagem atual	Recuperação: busca e seleciona trechos com procedência em uma coleção.
Realizar uma operação verificável	Ferramenta: calcula, lê dados estruturados ou executa uma operação autorizada.
Continuar sem perder o andamento	Estado e persistência: guardam resultados, decisões, pendências e registros.
Evitar ações indevidas e avaliar o resultado	Validação e revisão: aplicam restrições do software e decisões humanas.

O planejador e o coordenador não são a mesma coisa

Neste arranjo, o modelo pode propor uma sequência sem controlar diretamente a máquina. O coordenador é o código que decide se uma proposta é executável e autorizada. Um planejador pode ser o próprio LLM ou um componente adicional; não precisa existir como módulo separado em todo sistema.

O modelo não ganha uma ferramenta nova porque inventou seu nome. A aplicação precisa oferecer essa operação, implementar seu comportamento e permitir seu uso.

2 • DO TEXTO AO CICLO

Uma consulta pode exigir várias voltas

A primeira resposta do modelo pode não ser a resposta ao usuário. Ela pode solicitar uma busca ou um cálculo. Depois da execução, o modelo precisa receber a observação para decidir se já pode concluir ou se necessita de outra operação. [T2]

1. A pessoa formula um objetivo. O coordenador prepara a mensagem com as informações disponíveis e as ações permitidas.
2. O modelo interpreta a mensagem e propõe uma ação estruturada.
3. O coordenador verifica a proposta. Uma proposta recusada não deve ser apresentada como executada.
4. Uma ferramenta aceita recebe os argumentos e devolve dados ou um erro.
5. O coordenador registra a observação e a inclui no próximo contexto enviado ao modelo.
6. O modelo usa o resultado para propor outra operação ou redigir a resposta final. O ciclo também pode parar por erro, limite de passos ou decisão humana.

Resultado intermediário não é resposta final

Recuperar quatro fontes significa que o buscador terminou uma operação. Elas podem ser fichas curadas, notas locais ou fragmentos do corpus RAG. Isso não significa que o modelo recebeu o livro inteiro nem que já respondeu à pergunta. Da mesma forma, obter três ângulos é uma saída de cálculo; explicá-los à pessoa é outra etapa.

Onde entra o mensageiro?

Imagine um general fora do quartel: recebe relatórios por um mensageiro, devolve uma orientação, a equipe executa e envia um novo relatório. Na analogia, o LLM propõe; o aluno transporta; o coordenador verifica; as ferramentas executam. O modelo não é uma autoridade infalível: o usuário e as regras do aplicativo conservam o controle.

Na ponte manual, copiar e colar realiza o transporte que uma integração programática poderia automatizar. A necessidade do mensageiro decorre da ausência dessa conexão, não simplesmente de o modelo ser remoto. Um modelo remoto pode estar integrado; um modelo local também precisa de uma interface de comunicação.

Não há um número universal de voltas. Uma pergunta pode encerrar rapidamente; outra exige busca, leitura de dados, cálculo e revisão. A aplicação precisa estabelecer condições de parada.

3 • O CONTRATO

JSON descreve uma proposta; não a executa

JSON é um formato de intercâmbio de dados: objetos contêm pares de nomes e valores. Aspas duplas delimitam textos; chaves delimitam objetos. O formato não define o significado de `buscar_fontes` ou `finalizar`: isso pertence ao contrato do aplicativo. [T3]

```
{
  "acao": "buscar_fontes",
  "argumentos": {
    "consulta": "personagens principais papel"
  }
}
```

Parte	Leitura do exemplo
acao	Solicitar a operação de busca cujo nome é <code>buscar_fontes</code> .
argumentos	Entregar os parâmetros exigidos por essa operação.
consulta	Usar estes termos na busca; não é uma lista de resultados.

Três mensagens diferentes

O contexto de ida inclui instruções, objetivo, ações permitidas e observações; pode conter JSON, mas não é necessariamente um único objeto JSON. A proposta de volta obedece ao contrato da ação. A observação é o retorno da função realmente executada. Confundir esses três elementos pode produzir repetições ou respostas sem fundamento.

Validações antes de executar

Primeiro, o texto precisa ser JSON válido. Depois, a ação precisa existir, os campos precisam corresponder ao contrato e os dados devem respeitar os limites da operação. Por fim, é necessário verificar se a execução é permitida naquele estado. Formato correto não garante uma decisão apropriada.

Sublinhados não precisam de escape: escreva `buscar_fontes`, sem uma barra antes de `_`. Para representar uma aspa dupla dentro de um texto JSON, use o escape previsto pelo formato. Copie o objeto inteiro e mantenha todas as chaves de fechamento.

Um JSON de backup e um JSON de ação podem ser igualmente válidos como formato, mas não são intercambiáveis. Cada um tem seu próprio contrato.

4 • DIÁLOGO COMENTADO / IDA

Um pedido de alinhamento

Exemplo operacional: uma pessoa fornece dois pontos e pede verificar se o eixo, prolongado nos dois sentidos, alcança uma região-alvo já configurada. Não estudaremos aqui a origem histórica do destino; o foco é a colaboração entre componentes. [11]

Verifique o alinhamento nos dois sentidos.

A: latitude 51.17930555555555; longitude -1.8253083333333333.

B: latitude 51.17882972222222; longitude -1.8262194444444444.

O contexto informa ao modelo que `calcular_novo_alinhamento` recebe um nome e os pontos a e b, cada qual com lat e lon. As coordenadas devem ser copiadas literalmente como texto; conversões e cálculos pertencem ao programa.

Proposta que o modelo poderia devolver

```
{
  "acao": "calcular_novo_alinhamento",
  "argumentos": {
    "nome": "Exemplo didático",
    "a": {
      "lat": "51.17930555555555",
      "lon": "-1.8253083333333333"
    },
    "b": {
      "lat": "51.17882972222222",
      "lon": "-1.8262194444444444"
    }
  }
}
```

Leitura: “solicito `calcular_novo_alinhamento` com estes dois pontos”. O objeto não contém a conclusão de alinhamento. O modelo não declara que calculou nem substitui os valores fornecidos por medições de um catálogo.

Este JSON é um exemplo didático controlado, não o registro de uma consulta a um modelo durante a preparação do guia. Seus argumentos foram usados para testar a ferramenta de cálculo.

A ausência de um argumento de destino é deliberada: neste contrato, a região-alvo é definida pela aplicação. A IA não pode mudar silenciosamente a área para obter outro resultado.

O programa produz a observação

Ao receber a proposta, o coordenador verifica a estrutura e encaminha a ação ao módulo de cálculo. A validação confere formatos, faixas permitidas e presença das quatro coordenadas na pergunta. Ela não verifica, por si só, se os pontos medidos representam o eixo físico correto. [11]

O programa converte as coordenadas para números, usa a região configurada e executa seus testes. Abaixo está um recorte fiel da saída obtida com os valores da página anterior; os demais campos foram omitidos somente para a leitura.

```
{
  "resultado": "cruza_interior",
  "eixo_cruza_regiao": true,
  "cruza_interior": true,
  "apenas_limite": false,
  "sentidos": {
    "A_para_B": {
      "cruza_regiao": true,
      "cruza_interior": true
    },
    "B_para_A": {
      "cruza_regiao": false,
      "cruza_interior": false
    }
  },
  "teste_original_planilha": {
    "A_para_B": {
      "angulos": {
        "alpha": 27.61794861983585,
        "beta": 27.548971757013216,
        "gamma": 27.57608895572885
      }
    }
  }
}
```

A observação completa também conserva os pontos originais, os valores convertidos, o destino e os limites do método. Na rodada, o coordenador acrescenta o identificador CALC-NOVO-1 e a localização “Resultado da ferramenta nesta execução, etapa 1”.

O que esses campos autorizam concluir?

O eixo atravessa o interior da região pelo sentido A para B. O sentido B para A não a alcança. A análise trata longitude e latitude como coordenadas de um plano. Ela não é uma verificação geodésica e não determina a intenção de quem construiu uma avenida.

A observação precisa voltar ao modelo. Mostrar o resultado na tela da pessoa não o torna automaticamente visível ao LLM que está em outra aplicação.

Interpretar sem alterar os números

Depois de receber a observação, o modelo pode concluir. No contrato didático, a mensagem abaixo pede ao coordenador que apresente uma resposta final. Ela não solicita um segundo cálculo.

```
{
  "acao": "finalizar",
  "argumentos": {
    "texto": "O eixo cruza o interior da região-alvo no sentido A para B. No sentido B para A, não cruza. O ângulo gama do teste ABS é 27.57608895572885 graus. O resultado usa o plano longitude/latitude; não é um teste geodésico nem prova de intenção histórica. Fonte: CALC-NOVO-1, resultado da ferramenta nesta execução, etapa 1."
  }
}
```

Por que a resposta é diferente da primeira proposta?

Na primeira volta, faltava o resultado: o modelo escolheu uma operação. Na segunda, já há uma observação: ele pode explicar seu significado. O coordenador recebe finalizar, encerra o ciclo e mostra argumentos.texto. A referência identifica a saída do programa, não um documento que o modelo supostamente teria lido.

Dois testes não precisam significar a mesma coisa

Neste exemplo, o módulo também preserva um critério angular com valores absolutos e um teste de interseção que distingue as direções. O teste ABS devolve verdadeiro para o intervalo em ambas as orientações; a interseção direcional encontra a região somente de A para B. Não há contradição sobre o eixo: os testes preservam informações diferentes. [11]

O LLM deve manter essa diferença e não transformar desvio angular em incerteza de medição. Se uma execução devolver apenas um erro, a resposta precisa dizer que o cálculo não foi concluído, e não inventar números ou atribuir o erro ao usuário sem investigação.

Pergunta de leitura: quem produziu as coordenadas, quem selecionou a ação, quem fez a conta, quem transportou os resultados e quem explicou a conclusão? São responsabilidades distintas.

O mesmo padrão pode ser usado para ler uma tabela, verificar uma unidade, consultar um inventário ou preparar um rascunho. O instrumento muda; a separação entre proposta, execução e observação permanece.

7 • RECUPERAÇÃO E RAG

Dar ao modelo o material de que ele precisa

Uma pergunta sobre uma coleção particular não deve ser respondida apenas com associações gerais do modelo. Um recuperador seleciona trechos da coleção e os entrega como contexto; depois, o LLM gera a resposta usando esses trechos. Essa combinação é chamada de geração aumentada por recuperação, ou RAG (Retrieval-Augmented Generation). RAG não exige, por definição, embeddings: a recuperação pode ser lexical, vetorial ou híbrida. [T4]

```
Pergunta
  ↓
LLM propõe buscar_fontes
  ↓
Coordenador → RAG local lexical
  ↓
fichas + fragmentos de W → até 4 resultados
  ↓
observação volta ao LLM → resposta fundamentada
```

RAG não é treinar o modelo nem enviar o livro inteiro

Na edição 6.4, a recuperação combina 71 fichas curadas com 130 fragmentos de duas páginas que cobrem as 260 páginas de W. O PDF original não é enviado ao LLM: a busca seleciona, no máximo, quatro resultados por chamada e somente esses resultados entram no contexto transportado. Acrescentar esse corpus não altera os pesos do Transformer.

Nesta implementação, a recuperação é lexical e executada no navegador; não há embeddings nem banco vetorial. As fichas dão sínteses de alto nível e os fragmentos W-RAG funcionam como respaldo para detalhes não previstos nas fichas. A etapa de geração continua no LLM externo. Isso ilustra RAG sem reproduzir o recuperador denso nem o treinamento do artigo original. [T4, I1]

Exemplo: quando uma ficha não basta

Considere a pergunta: “Quais são as dimensões da Atlântida na novela W?”. Uma ficha temática pode não conter esse detalhe. O modelo pode então solicitar uma busca mais específica no corpus textual.

```
{
  "acao": "buscar_fontes",
  "argumentos": {
    "consulta": "Atlântida dimensões 3000 2000 estádios estádio egípcio"
  }
}
```

Uma recuperação bem-sucedida pode devolver W-RAG-083-084, localizado em W, cap. 35, pp. 83–84. Nesse fragmento aparecem a planície oblonga de 3.000 × 2.000 estádios, a escolha narrativa do estádio egípcio de 209,2 m e a conversão para 627,6 × 418,4 km. O LLM recebe esse trecho e suas páginas; não recebe automaticamente as outras 258 páginas.

```
{
  "fontes": [
    {
      "id": "W-RAG-083-084",
      "localizacao": "W, cap. 35, pp. 83–84",
      "texto": "... 3000×2000 estádios ... 209,2 m ... 627,6×418,4 km ..."
    }
  ]
}
```

Esse segundo JSON é uma observação produzida pelo recuperador, não uma proposta do modelo. Na próxima volta, o modelo pode usar a evidência para responder e citar o fragmento. Esta separação entre decisão, recuperação e geração é o ponto central do RAG agêntico.

Procedência e limites

Uma ficha de leitura é uma síntese curada; um resultado W-RAG é um fragmento textual extraído do PDF, com páginas. Nenhum deles transforma uma fala de personagem em medição científica. Da mesma forma, uma referência bibliográfica não significa que o relatório citado foi consultado. Um sistema útil conserva essas diferenças nos metadados e na resposta.

Nenhum resultado encontrado significa “esta busca não recuperou informação entre os resultados selecionados”, não “a informação não existe no livro”. Termos, idioma, pontuação lexical e o limite de quatro retornos podem explicar a lacuna; refine a consulta antes de concluir ausência.

A qualidade do sistema depende também da organização da base: nomes desambiguados, páginas, versões, fragmentação e metadados fiéis. As fichas e o corpus completo cumprem papéis complementares: uma interface bonita não compensa recuperação ruim ou atribuição incorreta.

8 • ESTADO E MEMÓRIA

O que fica guardado e o que chega à rede

Elemento	O que é e o que não é
Biblioteca	Coleção consultável de documentos ou fichas. Não é automaticamente o contexto de cada chamada.
Contexto da chamada	Mensagens e dados que o modelo recebe agora. Pode conter uma seleção do histórico e da biblioteca.
Estado da execução	Objetivo, etapa atual, resultados, erros e pendências. Permite decidir o próximo passo e evitar repetição.
Persistência	Gravação do estado para uso posterior. É diferente de manter uma tarefa executando.
Pesos do modelo	Parâmetros aprendidos no treinamento. Guardar uma nota no aplicativo não os modifica.
Registro de eventos	Evidência do que foi solicitado, aceito, executado e devolvido. Não é acesso ao raciocínio privado da rede.

Uma aplicação não é “a memória inteira da IA”

O coordenador pode guardar muitos eventos e enviar somente alguns ao modelo. Isso reduz tamanho, mas pode perder detalhes se a seleção for mal feita. Reenviar o contexto atualizado é essencial quando o serviço externo não vê diretamente o estado do aplicativo.

Local, remoto e persistente

Armazenamento no navegador depende do aparelho, do navegador e da origem do site. Exportar um projeto cria uma cópia transportável; não sincroniza automaticamente outros usuários. Na implementação didática, fechar a página interrompe o trabalho, mesmo que o site continue publicado. [I1]

Contexto, arquivo salvo, histórico e parâmetros treinados não são sinônimos. Antes de dizer “a IA lembra”, identifique onde a informação está armazenada e como será recuperada.

9 • LIMITES E CONTROLE

Ferramentas úteis exigem limites explícitos

O LLM produz uma proposta que pode estar errada, incompleta ou fora do contrato. O coordenador não deve executar texto arbitrário como código. Deve aceitar apenas operações conhecidas com argumentos apropriados. A existência de uma mensagem bem escrita não cria autorização. [T2, T3]

Situação	Conduta esperada
JSON truncado ou campos inesperados	Recusar a estrutura e indicar o problema; não completar silenciosamente valores importantes.
Coordenadas incompletas ou alteradas	Solicitar correção; não buscar um monumento parecido para substituir a entrada.
Ferramenta devolve erro	Registrar e devolver o erro ao modelo; distinguir falha de ausência de resultado.
Documento contém uma ordem para o agente	Tratar o documento como dado; suas frases não passam a substituir as instruções e permissões do aplicativo.
Ação repetida ou ciclo sem progresso	Usar histórico e limites de passos; encerrar ou pedir revisão humana.
Texto a publicar ou mudança irreversível	Exigir a autorização apropriada. Criar rascunho não é publicar.

A pessoa participa de mais de uma maneira

Ela define o objetivo e julga a utilidade do resultado. Na ponte manual também é mensageira. Transportar mensagens não a obriga a alterar fórmulas ou inventar argumentos; a tarefa é levar o conteúdo completo e verificar se o próximo resultado corresponde ao pedido.

Automatização não elimina a revisão

Uma integração por API poderia automatizar a transmissão. Isso exige implementar a conexão e controlar autenticação, permissões, custos e tratamento de falhas. Automatizar o transporte não torna os dados verdadeiros nem torna desnecessário revisar uma conclusão.

A segurança não depende de pedir educadamente ao modelo que se comporte bem. Depende também do que o software permite executar e de quais dados e recursos disponibiliza.

10 • O QUE TORNA O SISTEMA AGÊNTICO?

Escolher ações, observar e prosseguir

Há um contínuo entre respostas textuais, fluxos com ferramentas e agentes. Um fluxo pode usar um LLM em uma etapa, mas manter a ordem das operações inteiramente programada. Em um arranjo agêntico, o modelo tem alguma margem para escolher ações e ajustar o percurso ao objetivo e às observações, sempre dentro das regras do aplicativo. [T1]

Arranjo	Quem determina o próximo passo?
Chat sem ferramentas integradas	O modelo produz texto; a pessoa decide o que fazer fora do chat.
Fluxo fixo	O código determina a sequência. Um LLM pode redigir uma parte, sem escolher o fluxo.
Fluxo adaptativo por regras	Condições programadas escolhem caminhos. Isso não prova que um LLM participou.
Ciclo agêntico com ferramentas	O modelo propõe ações a partir do objetivo e das observações; o coordenador controla a execução.
Vários agentes	Há mais de um componente com função de decisão/modelo. Não é o mesmo que ter várias funções auxiliares.

O exemplo manual é limitado, mas instrutivo

O modelo externo participa de verdade quando recebe o contexto e produz a proposta; o transporte é manual. Por isso a atividade não demonstra autonomia contínua: depende de intervenção humana a cada troca. Um JSON fornecido previamente pelo docente demonstra o contrato, não demonstra uma decisão feita naquele momento por um LLM.

Transfira para outro problema

Para organizar um TCC, o primeiro instrumento pode ser a recuperação de fontes. Para conferir medições, pode ser uma calculadora. Para acompanhar um projeto, estado e tarefas são relevantes. Comece com um produto verificável e acrescente somente os módulos que reduzem uma limitação concreta.

Síntese: o comportamento agêntico pertence ao sistema modelo + contexto + controle + ferramentas + retorno. Não é uma propriedade adquirida por um texto JSON isolado.

REFERÊNCIAS E LEITURA FINAL

Como conferir sua compreensão

1. Explique a diferença entre propor uma ação e executá-la.
2. Mostre onde uma saída de ferramenta entra no próximo contexto.
3. Identifique quem calcula os números e quem os interpreta no diálogo.
4. Explique por que acrescentar um corpus RAG não altera os pesos do modelo.
5. Descreva uma situação em que o coordenador deve recusar uma proposta.
6. Diferencie fluxo fixo, decisões por regras e escolhas de um LLM.
7. Explique o que a ponte manual demonstra e o que ela não demonstra.

Referências técnicas

[T1] Anthropic. Building effective agents (2024). Distinção entre fluxos programados e agentes; modelo ampliado por ferramentas, recuperação e memória. <https://www.anthropic.com/research/building-effective-agents>

[T2] OpenAI. Function calling. Documentação do ciclo em que o aplicativo executa a ferramenta e devolve sua saída ao modelo. <https://developers.openai.com/api/docs/guides/function-calling>

[T3] Bray, T. (ed.). RFC 8259: The JavaScript Object Notation (JSON) Data Interchange Format (2017). <https://www.rfc-editor.org/rfc/rfc8259.html>

[T4] Lewis, P. et al. Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks (2020). Artigo de referência; o exemplo manual deste material não reproduz o treinamento e o recuperador denso daquele trabalho. <https://arxiv.org/abs/2005.11401>

[I1] Implementação didática distribuída com este material, edição 6.4.0. Contratos das ações, RAG local lexical e saídas da calculadora. Os exemplos operacionais são específicos dessa implementação; não são comandos universais de um LLM.

Documentação técnica consultada em 22/10/2026. Este guia combina explicação conceitual com exemplos controlados da implementação didática. A edição 6.4 usa 71 fichas curadas e 130 fragmentos RAG de W; as saídas numéricas do exemplo foram produzidas pelo programa, não por uma execução de modelo comercial durante a preparação.

Estude o mecanismo geral; use o guia de atividades para observar uma implementação concreta. O critério é conseguir explicar responsabilidades e evidências, não memorizar nomes de ferramentas.