

BCGX Challenge

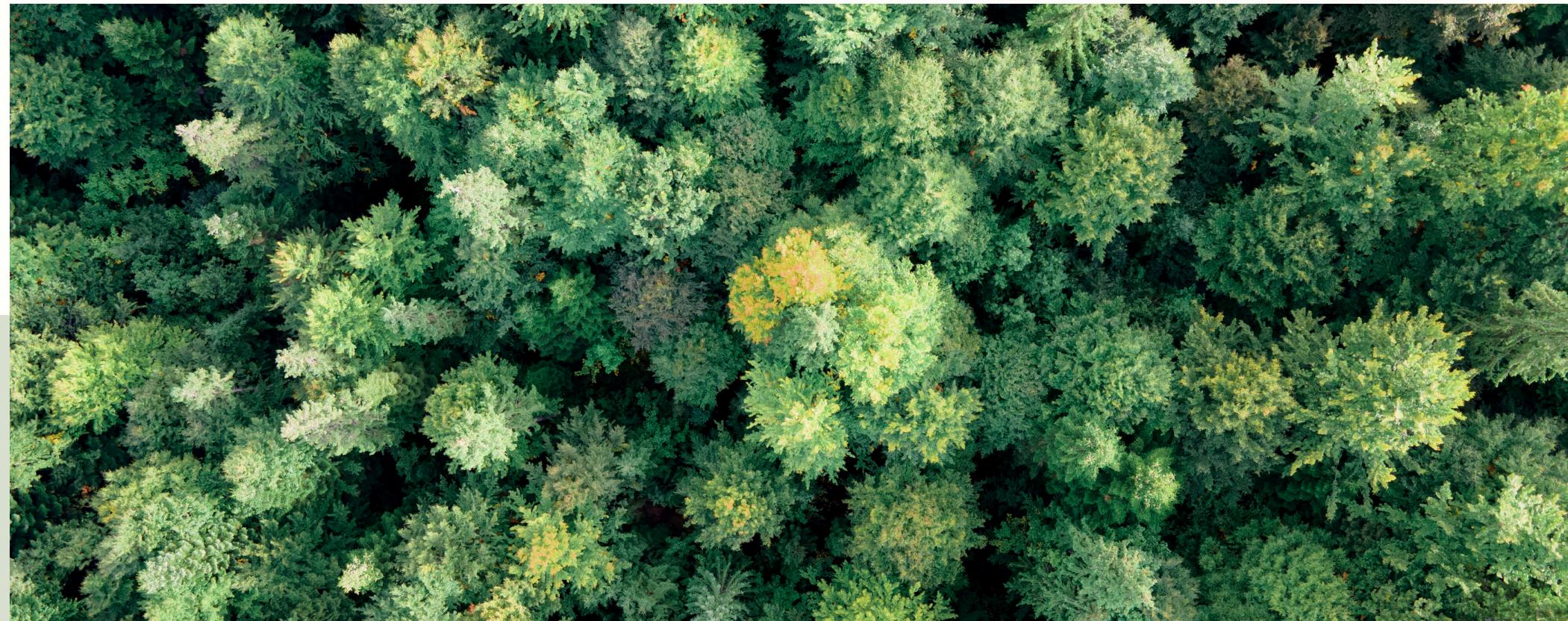
Assistente Climático Inteligente: IA no Combate às Mudanças Climáticas nas Cidades

Esdras Carvalho
Felipe Teixeira
Juliana Alves



Introdução

Neste projeto, criamos um chatbot com IA para ajudar gestores a implementar planos de gestão climática em suas cidades.



A nossa solução inclui:

- uma etapa de limpeza de dados;
- um método para criar chunks de textos baseados em semântica e, em seguida, salvar em um vectorDB;
- Uma complexa arquitetura de RAG em grafo para utilizar os textos ou dados da web na solução;
- Uma interface que, além de incluir o chat, traz highlights dos assuntos mais críticos para a cidade.

Processamento dos dados

Os dados foram estruturados, excluindo as páginas iniciais, como sumários e capas, e identificando a partir de qual página o documento efetivamente se inicia.

Utilizamos a biblioteca pdfminer para extrair o texto, removendo elementos indesejados, como números de página e repetições.

Chunking and Embedding

Criação dos chunks de texto de forma semantica

A partir dos dados tratados, geramos segmentos de texto para criar embeddings coerentes. Esses segmentos foram elaborados com base na semântica, garantindo que cada um aborde o mesmo tema e não confunda a LLM.

Embedding

O embedding foi feito utilizando o método generate_embeddings da openAi, utilizando o modelo de embedding text-embedding-ada-002 e em seguida salvo no vector DB (Para fins de otimização de tempo, salvamos os documentos "embeddados" em documentos .csv, assim para testes rápidos, ele cria o vector DB recebendo os dados destes docs)

Implementação do RAG

1. Início (__start__):

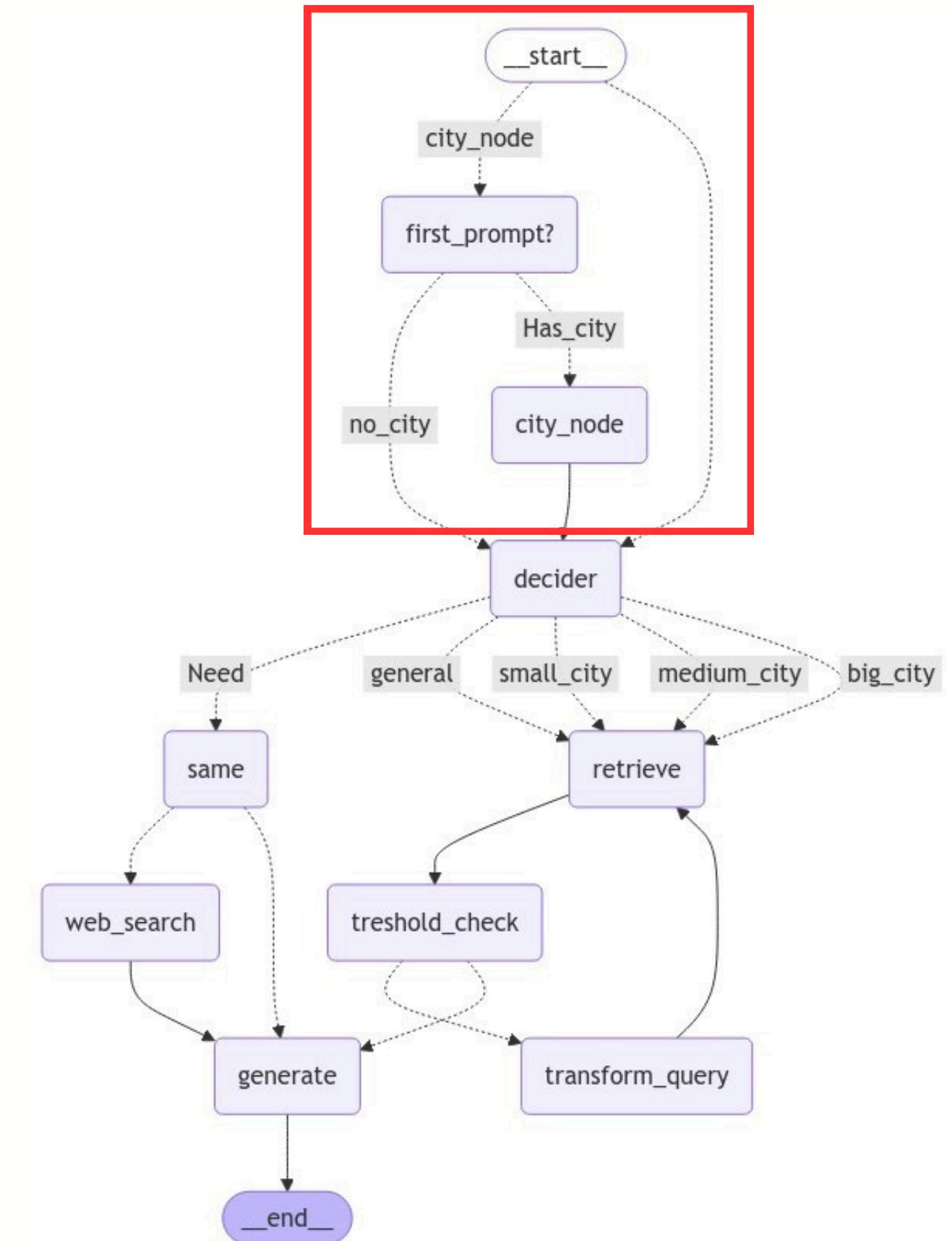
- Este nó inicia o chatbot a partir do prompt do usuário.

2. Nó de Cidade (city_node):

- O sistema verifica se a cidade foi fornecida no primeiro prompt (first_prompt?).
- Se a cidade não for fornecida no prompt (no_city), o sistema passa para o decider.

3. Decisor (decider):

- Após obter a cidade, caso seja o primeiro prompt, o sistema passa para o nó decider, ele é vira o primeiro node nos prompts subsequentes.
- O decider, numa outra chamada de llm, classifica o prompt em uma das seguintes categorias: Web_search, general, small_city, medium_city, big_city.



Implementação do RAG

4. Recuperação (retrieve):

- Através da categoria da prompt, o sistema pode recuperar informações específicas para inserir contexto para a LLM.

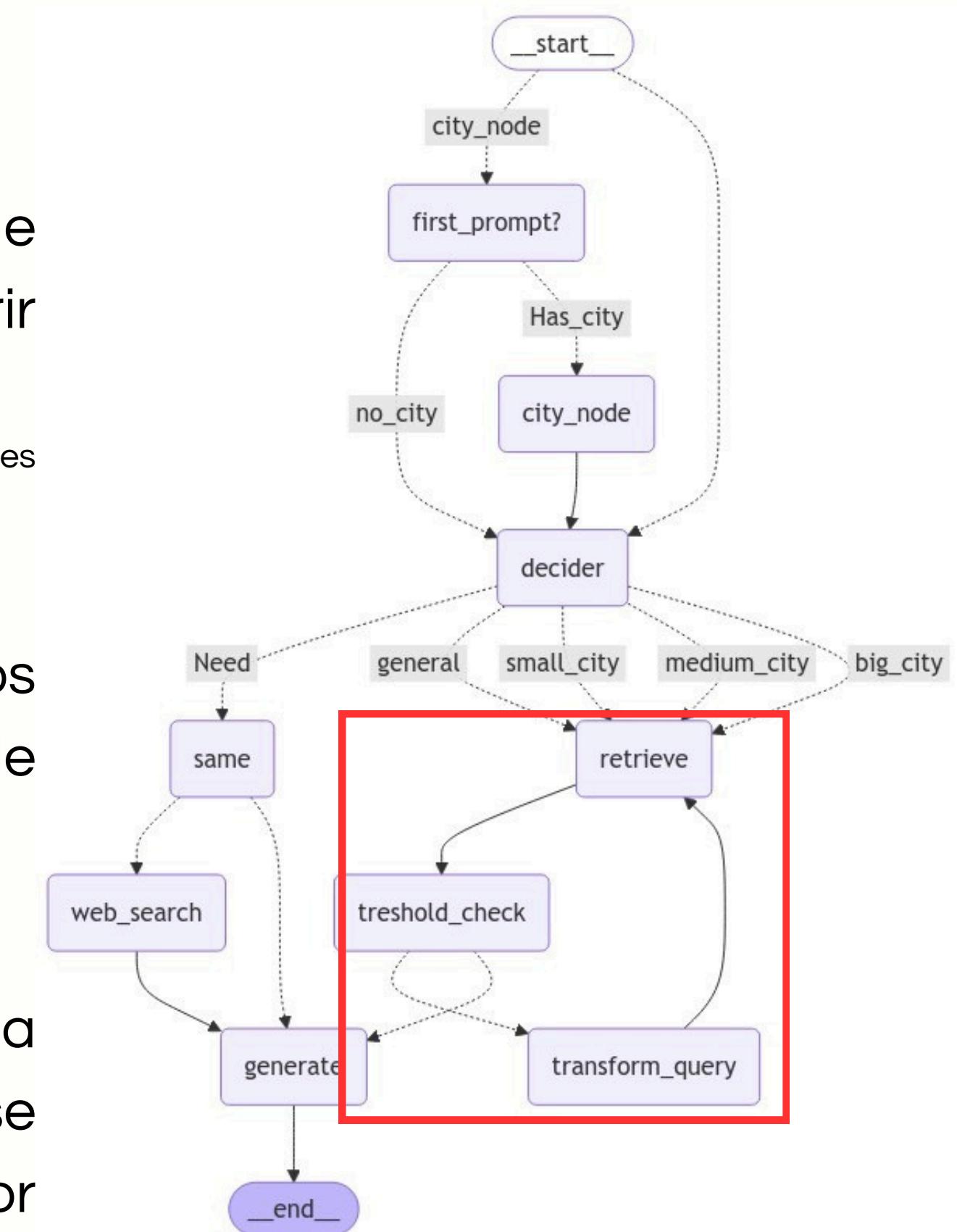
Lembrando que o “general” é sempre usado, apenas adicionamos ao retrieve certas informações dependendo do tamanho da cidade que o usuário escolhe

5.Verificação de Limite (threshold_check):

- Após a recuperação, o sistema verifica se os dados recuperados atendem a um limiar 0.8 pela função de similaridade de cossenos.

6. Transformação de Consulta (`transform_query`)

- Se necessário, o prompt é ajustado por uma chamada de `llm` para aumentar a precisão da recuperação. Esse processo pode ocorrer até 5 vezes; se o limite não for alcançado, a consulta é direcionada para gerar a resposta sem usar os `docs`, evitando um loop infinito.



Implementação do RAG

7. Pesquisa na Web (web_search):

- Caso o decider ache que o prompt do usuário não será respondido pelas informações do database ou não precisa de retrieve ele vai para esse caminho.
- O sistema decidirá se é preciso realizar uma pesquisa na web ou se a resposta pode vir direto, sem necessidade de contexto adicional..

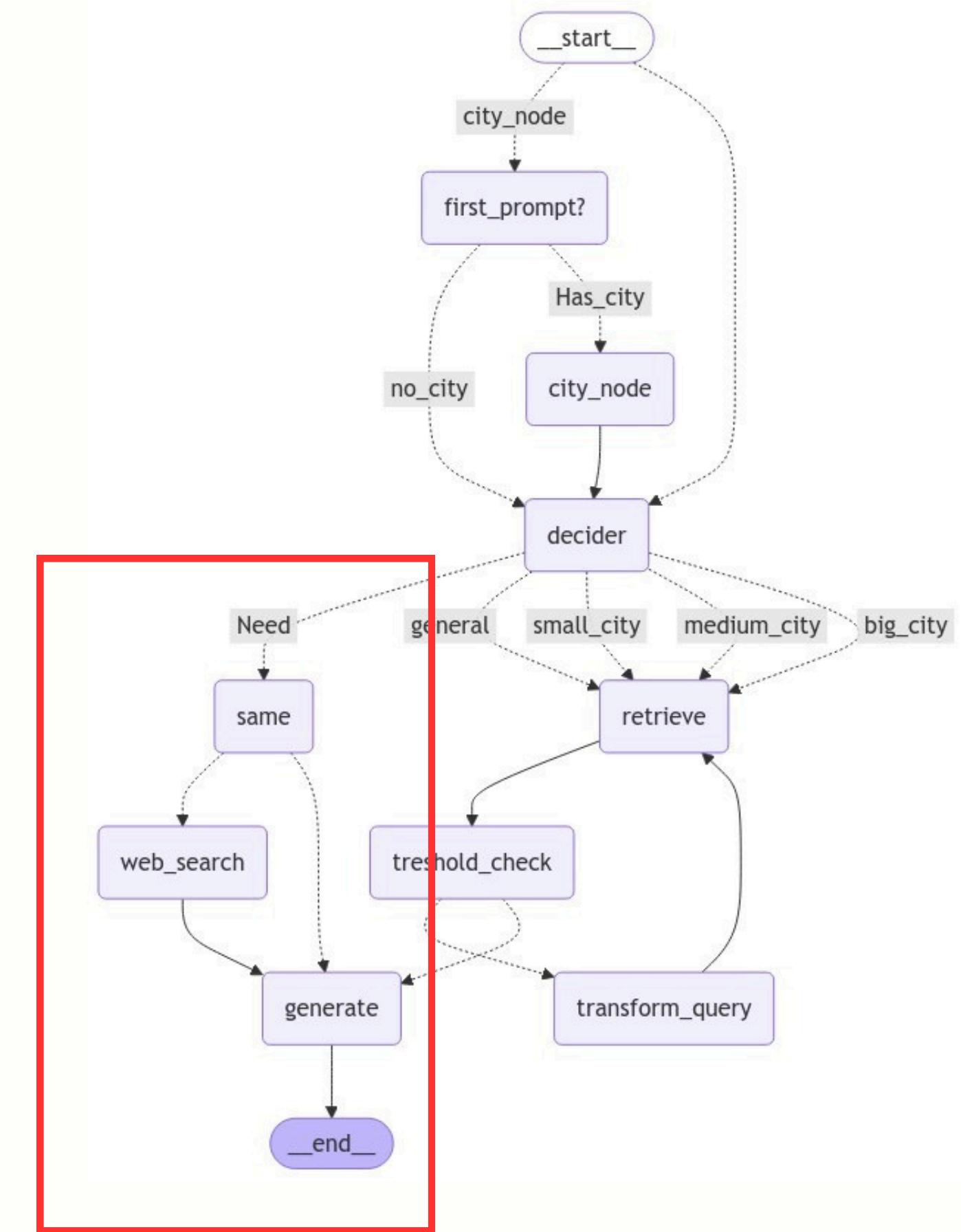
Nó **same** serve apenas para que o grafo funcione corretamente.

8. Geração (generate):

- O sistema gera a resposta final com base nas informações recuperadas e transformadas.

9. Fim (__end__):

- O processo termina após a geração da resposta final após o comando de terminar conversa.



Integração com Frontend

- Visualmente, além do chatbot, utilizamos uma barra lateral para ajudar o gestor a identificar os principais indicadores a serem monitorados. Isso aprimora sua experiência e o auxilia a formular perguntas de maneira mais direcionada.
 - Na perspectiva de negócios e agilidade, fornecer esses KPIs ao gestor pode ajudá-lo a direcionar melhor as perguntas sobre os principais problemas de sua cidade.
- Vale destacar que esses KPIs aparecem somente após o chatbot receber uma cidade válida e são atualizados apenas após o nó city_node, evitando que os KPIs mudem e confundam o gestor.

Balneário Barra do Sul

AI: Ola gestor climático da cidade de Balneário Barra do Sul. Como posso ajuda-lo hoje?

Você: Como posso começar meu plano?

AI: Para iniciar seu plano de planejamento climático, considere os seguintes passos:
Entenda o Contexto Local: Analise as características climáticas específicas de Balneário Barra do Sul-SC, incluindo dados históricos de clima, padrões de precipitação, temperaturas e eventos extremos.
Identifique os Riscos: Avalie os riscos climáticos que a cidade enfrenta, como aumento do nível do mar, erosão costeira, inundações e tempestades.
Engaje as Partes Interessadas: Envolve a comunidade local, especialistas e autoridades para obter uma visão abrangente e apoio no desenvolvimento do plano.
Defina Objetivos e Metas: Estabeleça objetivos claros para mitigar os riscos climáticos e adaptar a cidade às mudanças climáticas.
Desenvolva Estratégias: Crie estratégias de adaptação e mitigação que sejam viáveis e sustentáveis a longo prazo.
Monitore e Avalie: Implemente um sistema de monitoramento para avaliar a eficácia das estratégias e ajustar conforme necessário.
Esses passos ajudarão a estruturar seu plano de forma eficaz. Se precisar de mais detalhes sobre algum desses pontos, estou à disposição para ajudar!

Você: Como posso lidar com as inundações?

AI: Para lidar com as inundações em Balneário Barra do Sul-SC, considere as seguintes estratégias:
Infraestrutura de Drenagem: Melhore e amplie o sistema de drenagem urbana para...

Digite sua mensagem aqui...

Integração com Frontend

- E como fazemos para saber a cidade do usuário para podemos ver os principais problemas que o gestor enfrentará?
- Para evitarmos o problema de cidades digitadas de forma errada ou algo do tipo e para que o primeiro prompt sempre receba uma cidade, retornando suas características, adicionamos um botão no qual o gestor escolhe sua cidade antes de começar o chat, assim temos um primeiro prompt automático que receberá essa cidade e retornará informações sobre ela, logo o gestor já terá uma grande ajuda no início

Bem vindo ao assistente climático!

Para iniciar, por favor, selecione a UF e Município

Selecionar o estado

Selecionar o município

START CHAT

Bem vindo ao assistente climático!

Para iniciar, por favor, selecione a UF e Município

Selecionar o estado

Maranhão (MA)

Selecionar o município

Lagoa Grande do Maranhão

START CHAT

Conclusão e próximos passos



A implementação do "Climate Assistant" demonstrou ser uma solução eficaz para auxiliar gestores municipais na implementação de planos de gestão climática. Através de uma arquitetura complexa de RAG em grafo, o chatbot consegue fornecer informações precisas e relevantes, utilizando tanto dados internos quanto informações da web.

Próximos Passos:

1. Expansão da Base de Dados: Ampliar a base de dados interna com mais documentos relevantes para aumentar a precisão das respostas.
2. Aprimoramento do Algoritmo de Similaridade: Melhorar o algoritmo de similaridade para reduzir a necessidade de ajustes e consultas à web.
3. Integração com Novas Fontes de Dados: Incorporar novas fontes de dados externas para enriquecer as informações fornecidas pelo chatbot.
4. Melhoria da Interface: Refinar a interface do usuário, incluindo a barra lateral de KPIs, para tornar a interação mais intuitiva e eficiente.