



THE  
DEVELOPER'S  
CONFERENCE

# GraphRAG

Chega de mentiras, IA com evidências

Eliézer Zarpelão - Sr Solutions Engineer - Neo4J

# Agenda

1. **Grafos**
2. **LLMs**
3. **GraphRAG**
4. **Demo**



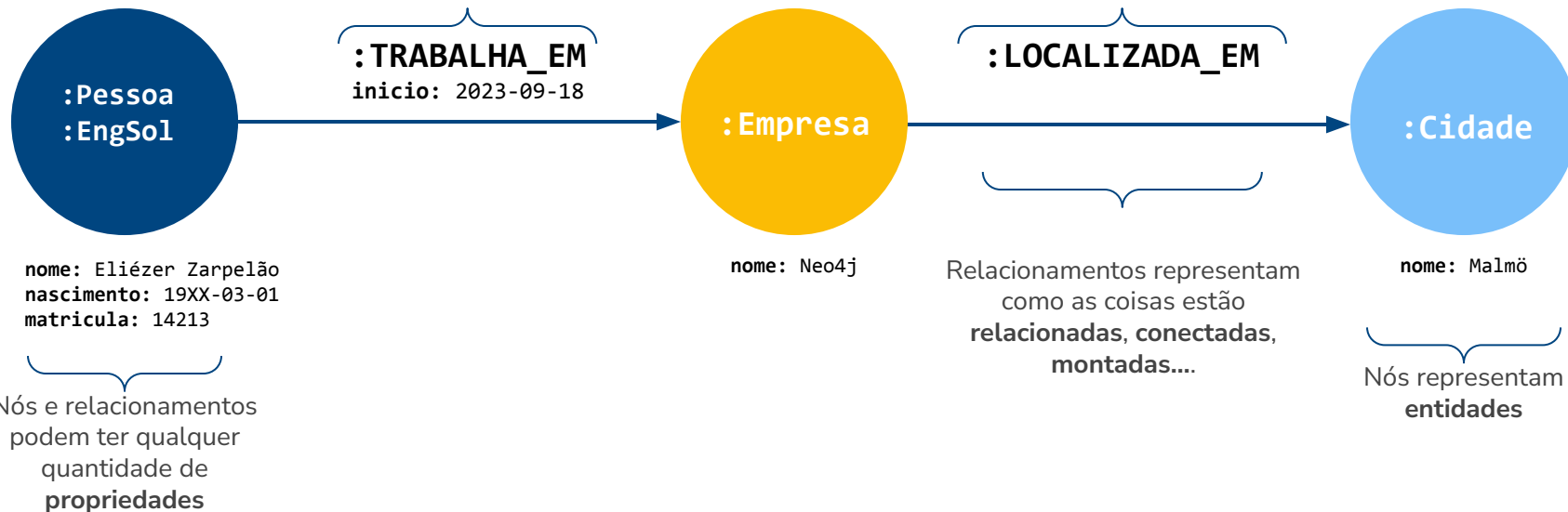
**Eliézer Zarpelão**  
Sr. Solutions Engineer

# O que são grafos?

Nós podem ter zero ou mais **rótulos/labels**

Relacionamentos tem **um tipo**

Relacionamento tem **direção**

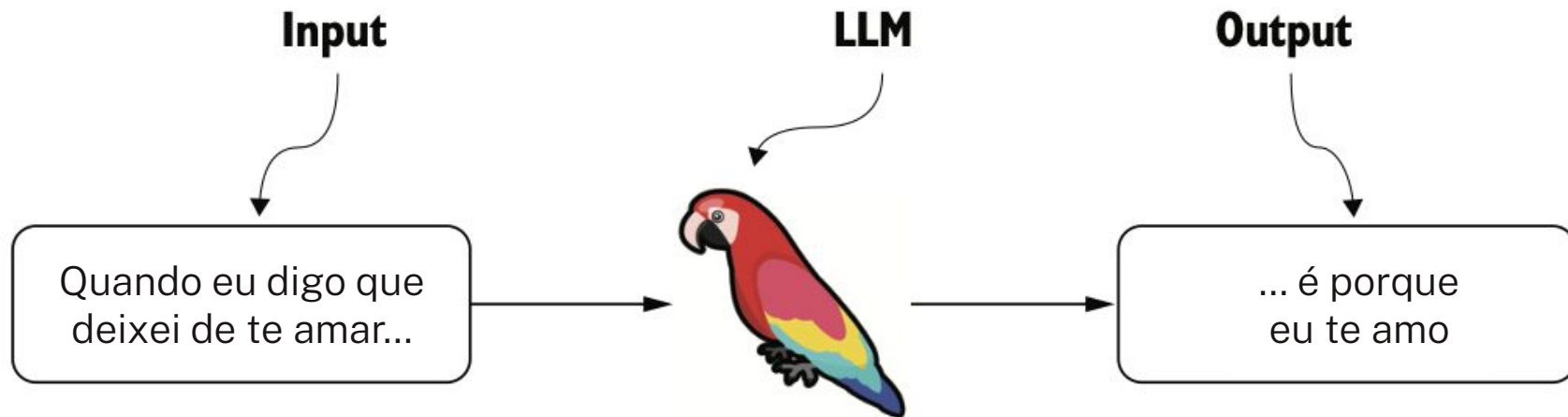




Quando eu digo que deixei de te amar...



# Como funciona as LLMs... 5M pés



# Limitações LLMs



## Knowledge Cutoff

Eventos posteriores ao último treino ficam fora do alcance

1

2

## Informações desatualizadas

Mesmo corretos no passado, respostas podem não refletir a realidade atual

## Alucinações

Podem inventar fatos/citações com segurança excessiva baseada em predição de textos

3

4

## Ausência de dados privados

Soluções exigem integração segura sem expor informações sensíveis

*Viés, falta de contexto, vulnerabilidade à prompt injection, respostas inconsistentes*

# Podemos resolver... com *supervised finetuning*



## Entrada

Quando eu digo que deixei  
de te amar

Quando eu digo que não  
quero mais você

E nessa loucura de dizer que  
não te quero



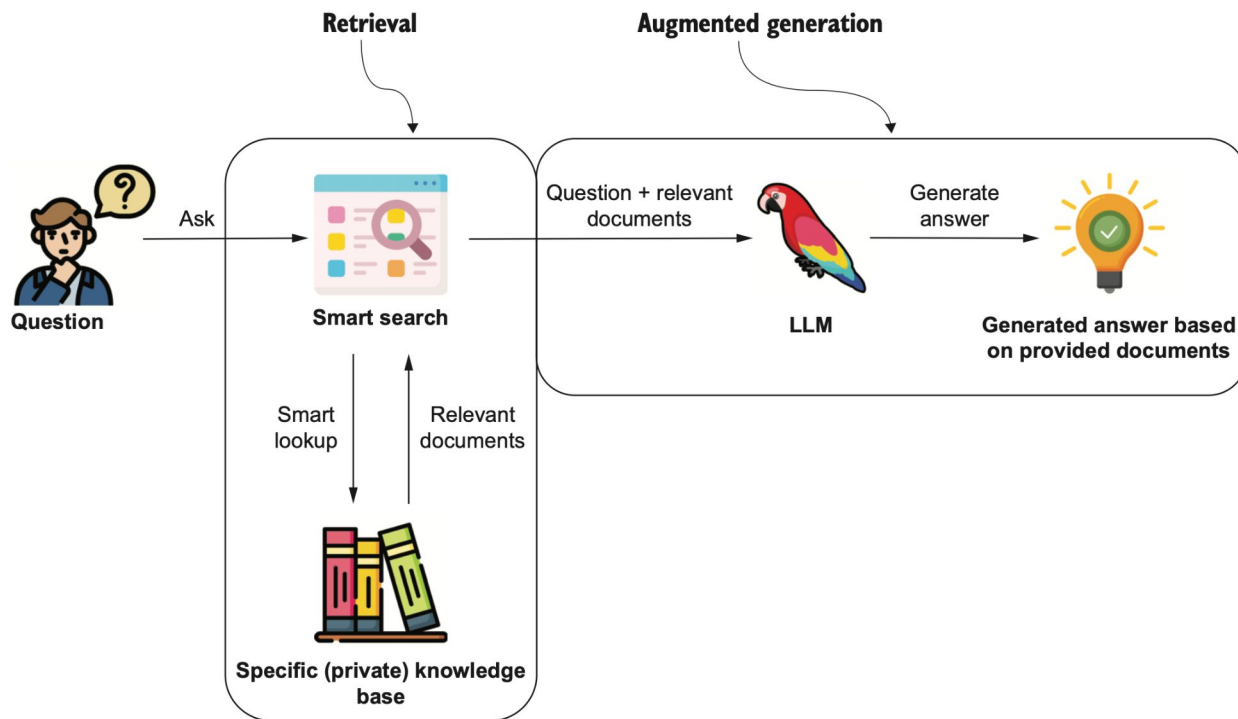
## Saída desejada

É porque eu te amo

É porque eu te quero

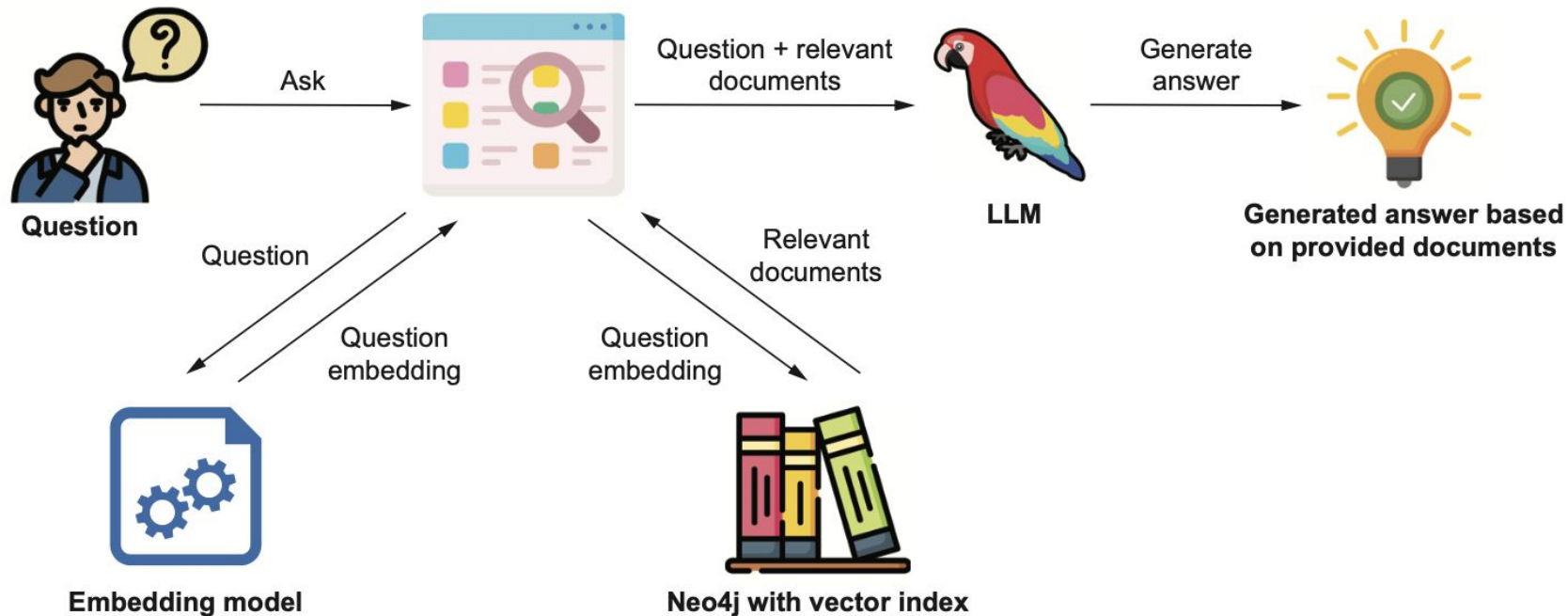
Vou negando as aparências,  
disfarçando as evidências

# Ou com retrieval-augmented generation (RAG)





# RAG usando busca de similaridade de vetores





# E o GraphRAG?

Calma, vamos construir nosso grafo de conhecimento primeiro...



# Construção do Knowledge Graph

## Dados não-estruturados

Normalmente PDFs ou outros documentos de texto



## Mix dados estruturados e não-estruturados

Dados estruturados com texto longo



## Dados estruturados

Dados estruturados com valores de texto curtos



# Construção do Knowledge Graph

## Dados não-estruturados

Normalmente PDFs ou outros documentos de texto

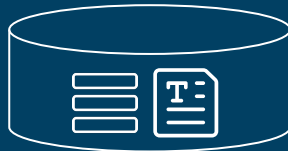
intrinsecamente **difícil**,  
ferramentas **imaturas**,  
Caso "hello world" 🤖



## Mix dados estruturados e não-estruturados

Dados estruturados com texto longo

**boa** metodologia,  
**boas** ferramentas,  
**maior parte** dos casos reais 👍



## Dados estruturados

Dados estruturados com valores de texto curtos

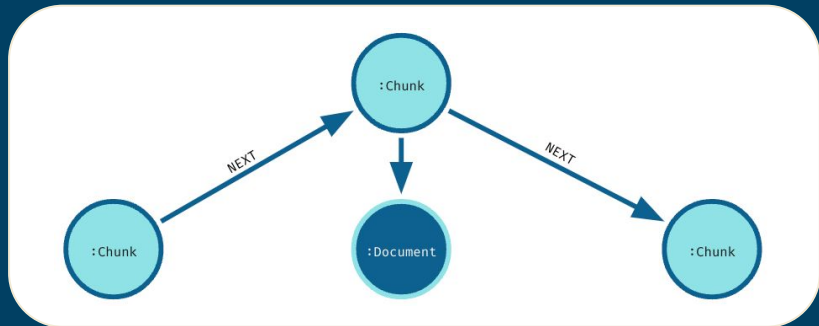
**boa** metodologia,  
**boas** ferramentas,  
**comum** nas empresas 👍



# Dois tipos de Knowledge Graphs

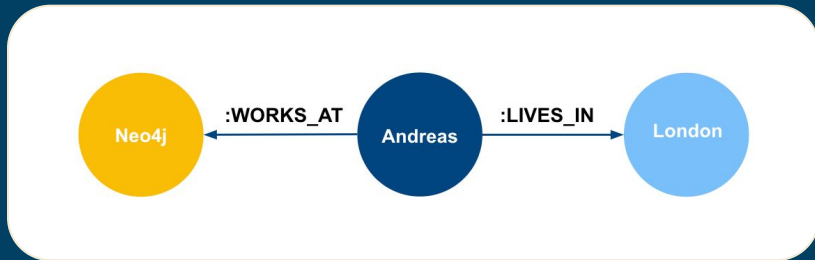
## Lexical Graph

Uma representação em grafo, por exemplo, de palavras, parágrafos, trechos, documentos e os relacionamentos entre eles



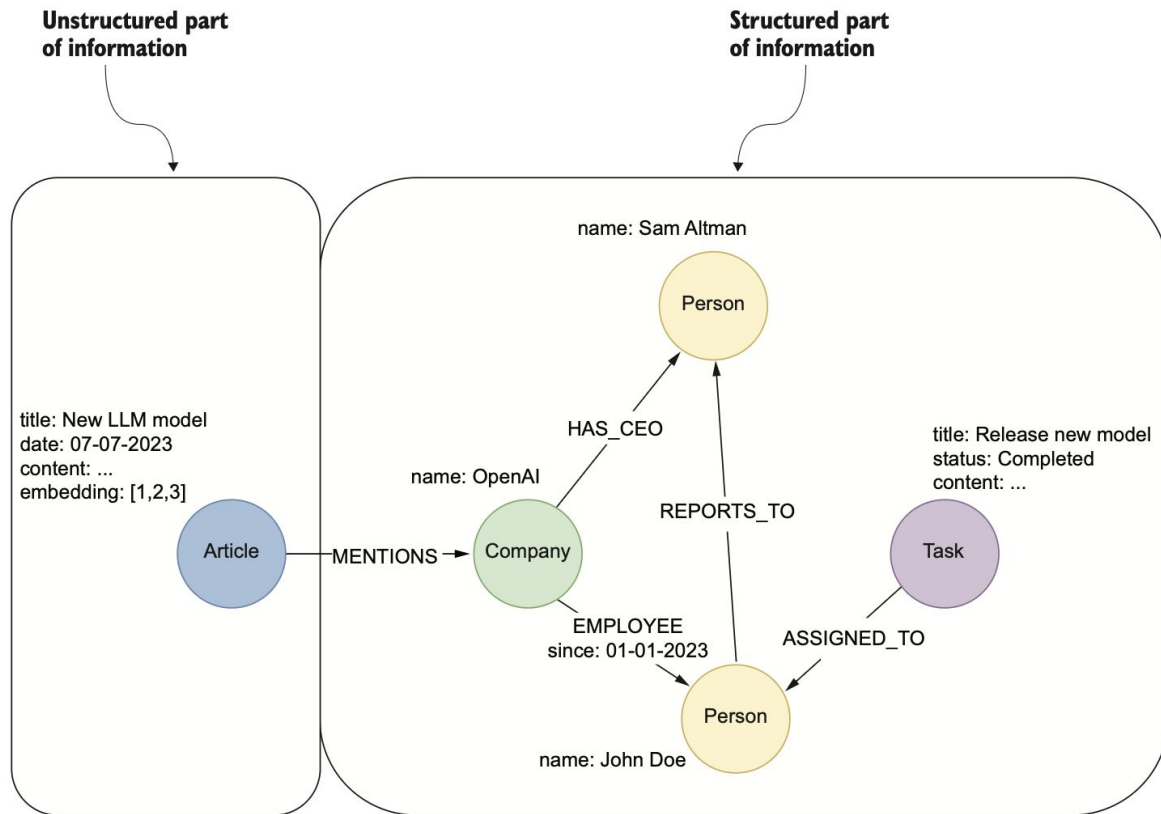
## Domain Graph

Uma representação em grafo do mundo real ou digital, ou seja, o domínio que você está modelando



**Esses dois não são exaustivos**  
**Também não são mutuamente exclusivos**

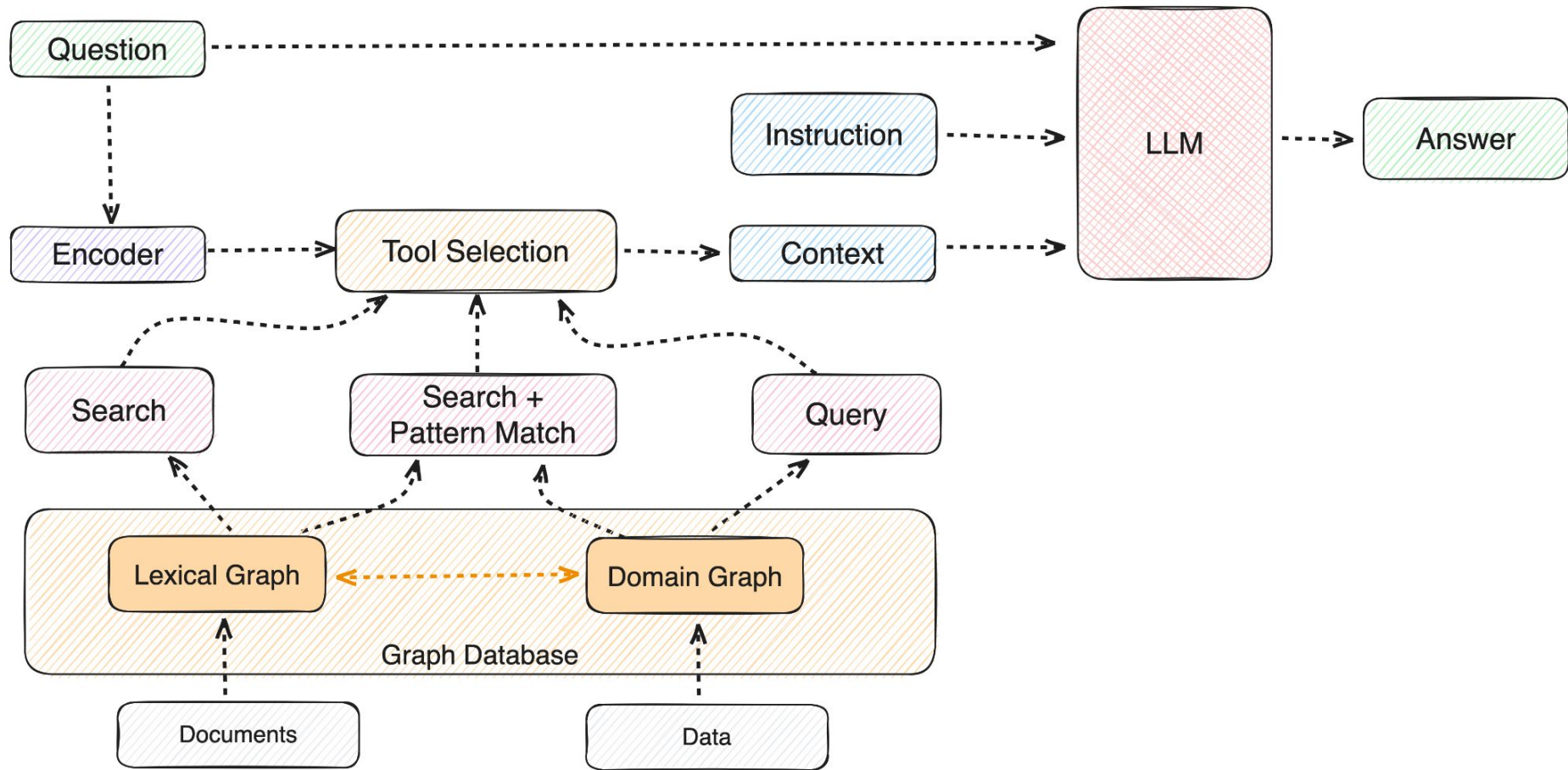
# Grafo de conhecimento como base do RAG



Maior **capacidade** de recuperar e organizar dados **contextualmente relevantes**, permitindo respostas:

- **precisas**
- **confiáveis**
- **explicáveis**

# Fluxo GraphRAG



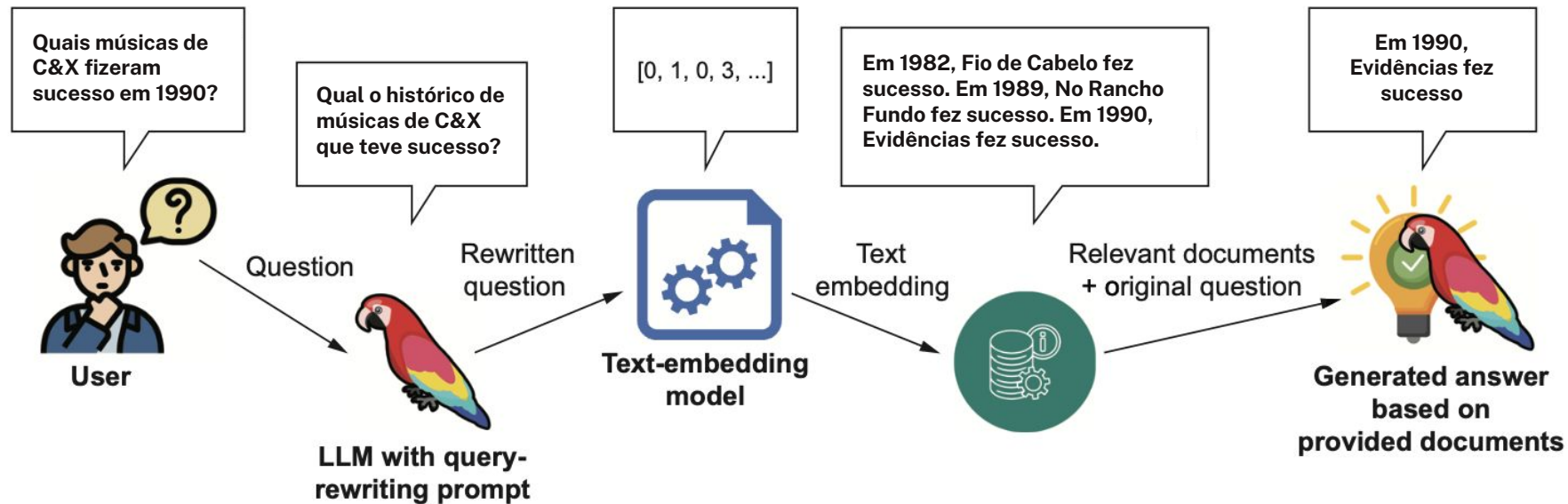
# Como melhorar a acurácia e recall?

*Step-back prompting* e outras  
estratégias de embeddings





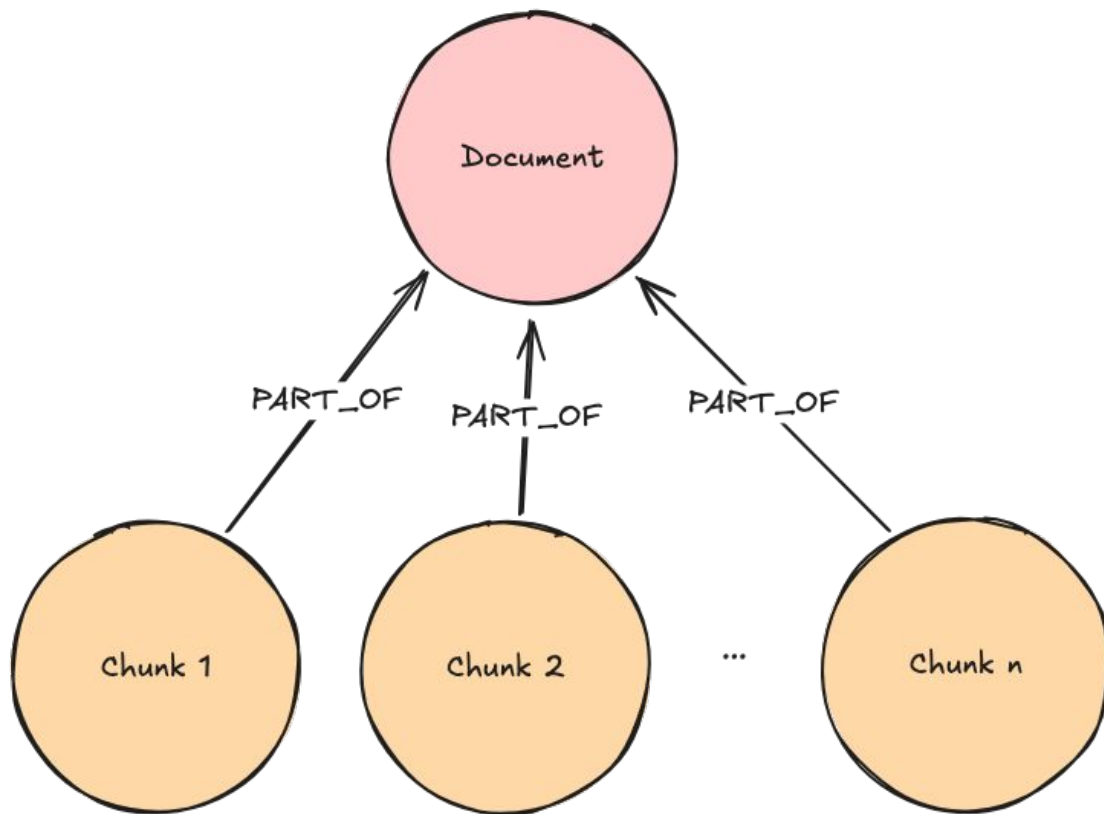
# Step-back prompting



# Basic retriever

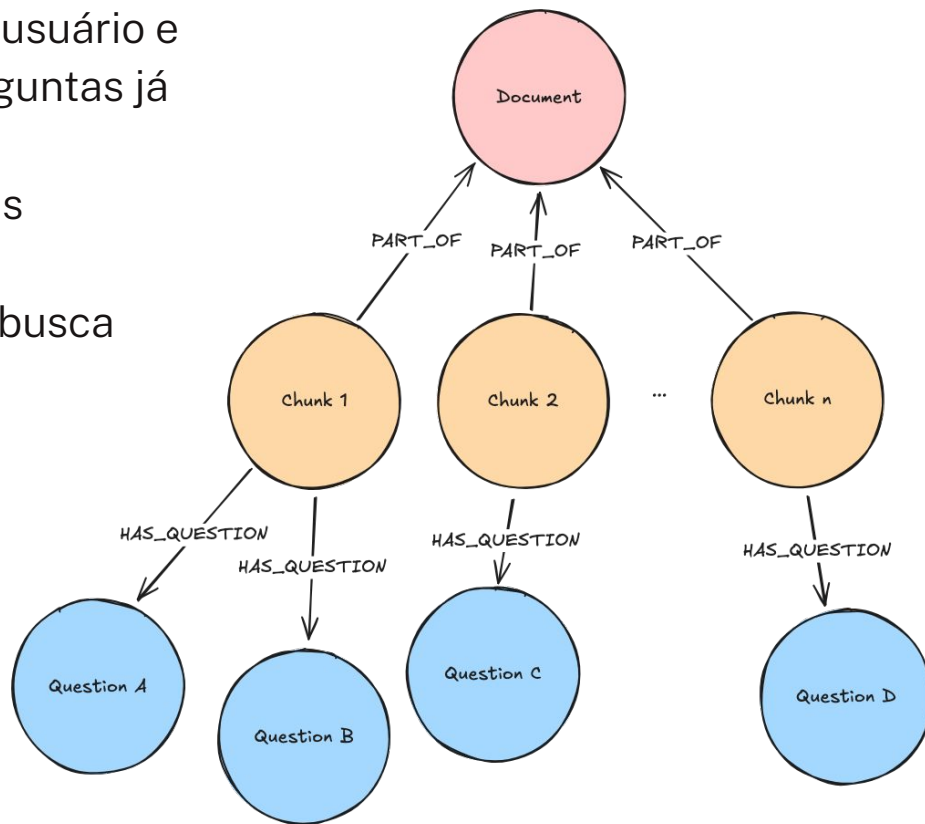
Out:

- Vector Retriever
- Naive Retriever
- Baseline RAG
- Basic RAG
- Typical RAG



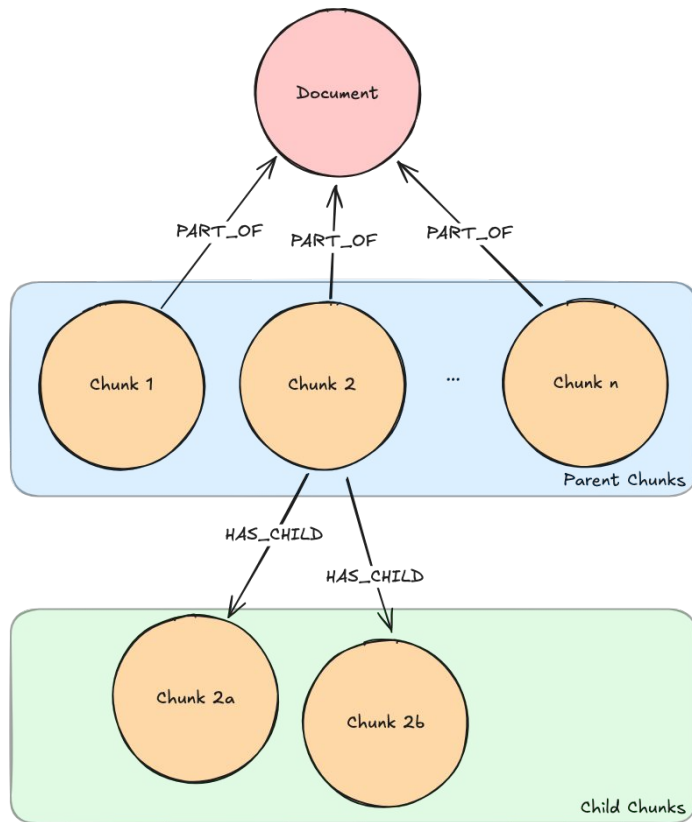
# Hypothetical Question Retriever

- Cria o *embedding* da pergunta do usuário e compara com *embeddings* de perguntas já geradas
- Recupera os chunks associados às perguntas mais semelhantes (k)
- Melhora resultados em relação à busca direta pergunta → texto
- **Exige pré-processamento**



# Parent-Child Retriever

- Chunks menores, vetores melhores
  - textos curtos trazem embeddings mais significativos
- Busca em child chunks
  - encontra trechos mais próximos da pergunta
- Recupera parent chunks
  - garante contexto mais amplo para resposta
- Pré-processamento
  - dividir em parent e child chunks, criar embedding apenas dos child

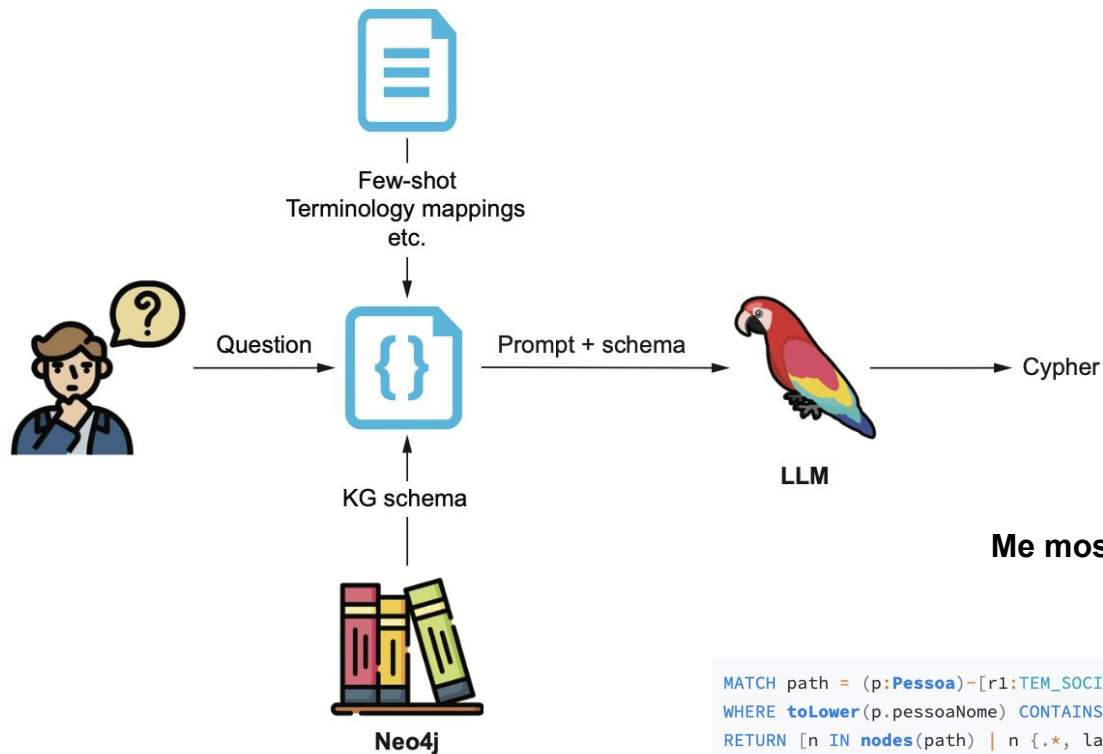


# Query language generation

Convertendo linguagem natural em  
**Cypher** (query language)



# Gerando Cypher a partir da questão

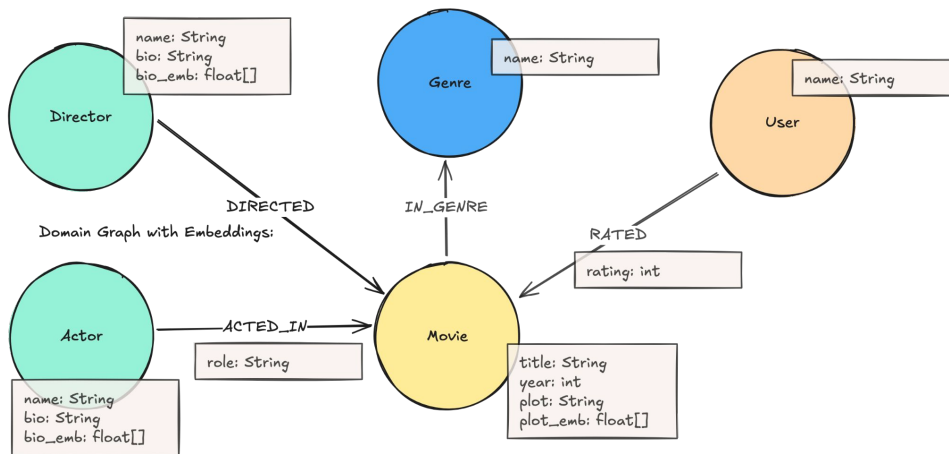


Me mostre em grafo o Luciano Huck e as empresas que ele tem sociedade.

```
MATCH path = (p:Pessoa)-[r1:TEM_SOCIEDADE_PESSOA]->(s:Sociedade)-[r2:ASSOCIADA_A_EMPRESA]->(e:Empresa)
WHERE toLower(p.pessoaNome) CONTAINS 'luciano huck'
RETURN [n IN nodes(path) | n {.*, labels: labels(n)}] AS nodes,
        relationships(path) AS relationships
```

# Cypher retriever

- Cypher Templates
  - Queries parametrizadas
- Dynamic Cypher Generation
  - Geração usando trechos pré-definidos
- Text2Cypher
  - Queries dinâmicas baseadas no schema



Cypher Templates



Dynamic Cypher Generation



Text2Cypher

# Benefícios do GraphRAG

**1. Maior  
precisão**

**2. Desenvolvimento  
facilitado**

**3. Explicabilidade  
& Governança**



# Grafos de conhecimento (KG) desbloqueiam GenAI

**KGs**

Conhecimento  
Fatos  
Contexto

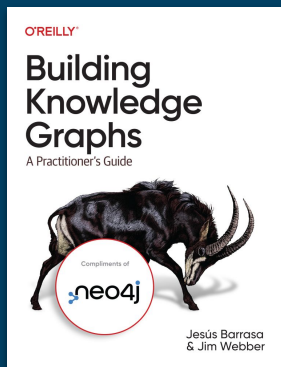


**GenAI**

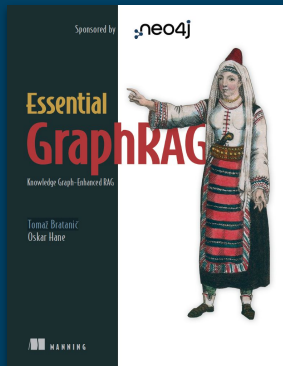
Linguagem  
Estatísticas  
Criatividade

Preciso  
Contextual  
Explicável

# Caminho das pedras com GraphRAG



<https://dev.neo4j.com/free-kg-book>



<https://neo4j.com/essential-graphrag/>



<https://www.deeplearning.ai/short-courses/knowledge-graphs-rag/>

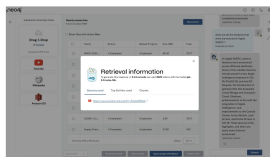


[neo4j.com/labs/genai-ecosystem/](https://neo4j.com/labs/genai-ecosystem/)



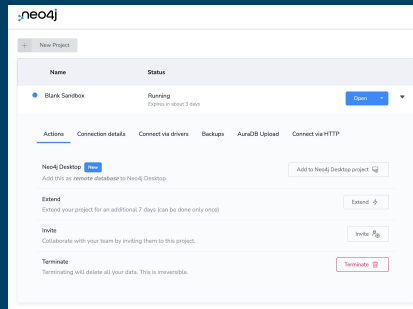
[discord.gg/graphrag](https://discord.gg/graphrag)

Neo4j LLM Knowledge Graph Builder -  
Extract Nodes and Relationships from  
Unstructured Text (PDF, YouTube,  
Webpages)

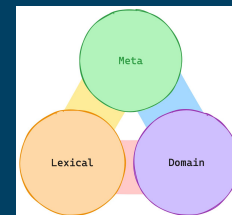


The Neo4j LLM Knowledge Graph Builder is an **online application** for turning unstructured text into a knowledge graph. It provides a magical text to graph experience.

<https://dev.neo4j.com/kg-builder>



<https://sandbox.neo4j.com/>



<https://graphrag.com/>



**KEEP  
CALM  
AND  
SHOW ME  
SOME MAGIC**

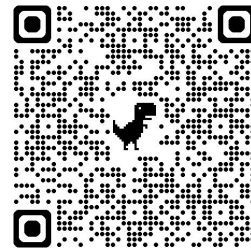
# Comece agora mesmo GRATUITAMENTE!

neo4j graphacademy



<https://graphacademy.neo4j.com/>

neo4j auraDB



<https://neo4j.com/cloud/platform/aura-graph-database/>



YouTube



Linkedin



Comunidade Neo4j

# Obrigado!!!