PONTIFÍCIA UNIVERSIDADE CATÓLICA DO RIO DE JANEIRO

Luíza Ferreira Camerini

PEPWatch: Base de Conhecimento de Pessoas Expostas Politicamente

PROJETO FINAL DE GRADUAÇÃO

Centro Técnico Científico - CTC

DEPARTAMENTO DE INFORMÁTICA

Programa de Graduação em Ciência da Computação

Rio de Janeiro
Novembro de 2025

Pontifícia Universidade Católica
do Rio de Janeiro

Luíza Ferreira Camerini

PEPWatch: Base de Conhecimento de Pessoas Expostas Politicamente

Relatório de Projeto Final apresentado como requisito parcial para obtenção do grau de Bacharelado pelo programa de Graduação em Ciência da Computação da PUC-Rio . Aprovada pela Comissão Examinadora abaixo:

Prof. Marcos Vianna Villas
Orientador
Departamento de Informática – PUC-Rio

Rio de Janeiro, 20 de Novembro de 2025

Luíza Ferreira Camerini

Ficha Catalográfica

Camerini, Luíza

PEPWatch: Base de Conhecimento de Pessoas Expostas Politicamente / Luíza Ferreira Camerini; orientador: Marcos Vianna Villas. – 2025.

56 f: il. color. ; 30 cm

Relatório de Projeto Final (graduação) - Pontifícia Universidade Católica do Rio de Janeiro, Departamento de Informática, 2025.

1. Informática – Projeto Final. 2. Grafo de conhecimento. 3. Pessoa Exposta Politicamente. 4. Base de conhecimento. I. Villas, Marcos. II. Pontifícia Universidade Católica do Rio de Janeiro. Departamento de Informática. III. Título.

CDD: 004

Agradecimentos

Ao meu orientador professor Marcos Villas pelo estímulo e parceria para a realização deste trabalho.

Ao pesquisador Bruno Bondarowski, que trouxe perguntas e perspectivas diversas sobre este trabalho.

À minha família, a qual possibilitou essa jornada profissional.

Finalmente, aos amigos que me apoiaram do início ao fim.

Resumo

Camerini, Luíza; Villas, Marcos. **PEPWatch: Base de Conhecimento de Pessoas Expostas Politicamente**. Rio de Janeiro, 2025. 57p. Relatório de Projeto Final de Graduação – Departamento de Informática, Pontifícia Universidade Católica do Rio de Janeiro.

Este documento apresenta a especificação, arquitetura e implementação de um sistema de identificação e análise de Pessoas Politicamente Expostas (PEP) desenvolvido sobre um banco de dados orientado a grafos Neo4j. O sistema integra dados de políticos brasileiros, seus parentes, sócios e atividades financeiras, permitindo a detecção de possíveis padrões de fraude, corrupção e nepotismo. Além disso, incorpora componentes de IA capazes de interpretar perguntas em linguagem natural e consultas Cypher, e retornar respostas contextualizadas e auditáveis. O objetivo principal é fornecer uma ferramenta robusta e transparente para apoiar processos de *due diligence*, investigações jornalísticas e análises de risco em conformidade com boas práticas de governança e *compliance*.

Palavras-chave

Grafo de conhecimento; Pessoa Exposta Politicamente; Base de conhecimento.

Abstract

Camerini, Luíza; Villas, Marcos (Advisor). **PEPWatch: Knowledge Base of Politically Exposed Persons**. Rio de Janeiro, 2025. 57p. Final Graduation Project Report – Departamento de Informática, Pontifícia Universidade Católica do Rio de Janeiro.

This document presents the specification, architecture, and implementation of a system for identifying and analyzing Politically Exposed Persons (PEPs) built on a Neo4j graph database. The system integrates data on Brazilian politicians, their relatives, business partners, and financial activities, enabling the detection of potential patterns of fraud, corruption, and nepotism. It also incorporates AI components capable of interpreting natural language queries and Cypher queries, and returning contextualized and auditable answers. The primary objective is to provide a robust and transparent tool to support due diligence processes, investigative journalism, and risk analysis in alignment with governance and compliance best practices.

Keywords

Knowledge Graph; Politically Exposed Person; Knowledge Base.

Sumário

Lista de Abreviaturas

PEP – Pessoa Exposta Politicamente

KB – *Knowledge Base*

RAG – *Retrieval-Augmented Generation*

API – *Application Programming Interface*

LN – Linguagem Natural

1
Introdução

Uma Pessoa Exposta Politicamente (PEP) é um agente presente no âmbito da política - como prefeito, presidente, deputado, etc. - ou um agente de relacionamento próximo com um político - como filho, pai ou mãe, sócio, amigo íntimo, etc. Relacionamentos entre PEPs podem abranger questões familiares, comerciais, sociais, etc. A existência e natureza destes relacionamentos é um assunto relevante, pois suas análises podem abrir caminhos investigativos contra atos de corrupção, fraude, terrorismo, entre outros atos ilícitos.

A ausência de uma base de conhecimento pública dificulta procuras e investigações mais organizadas e sistematizadas de possíveis atos ilícitos expostos por relacionamentos entre PEPs.

2
Situação Atual

Corporações possuem a necessidade indispensável de passar pelo processo de *due diligence*, ou seja, uma procura que revela fatores que sejam interessantes para a investigação profunda de uma empresa antes de uma movimentação financeira ou contrato. Plataformas como *AML Watcher* (WATCHER, 2025) e *OpenSanctions* (OPENSANCTIONS, 2025) são plataformas que disponibilizam dados sobre PEPs e suas empresas, transações comerciais, entre outros aspectos de interesse para *due diligence* e *compliance*.

Plataformas como *AML Watcher* e *OpenSanctions* desempenham um papel essencial no processo de *due diligence* ao fornecerem dados consolidados e atualizados sobre Pessoas Politicamente Expostas (PEPs), suas relações empresariais, sanções internacionais e atividades comerciais relevantes. O AML Watcher concentra-se no monitoramento de riscos relacionados à lavagem de dinheiro e financiamento ilícito, oferecendo listas de observação, relações societárias e alertas de conformidade. Já o OpenSanctions destaca-se por ser uma plataforma aberta que integra diversas bases globais de sanções, PEPs e entidades de interesse, permitindo consultas transparentes e reutilizáveis para investigações e auditorias.

Porém, não foram identificadas bases de conhecimento que abordem este tema e que sejam de acesso público e gratuito.

2.1
Principais conceitos

Nesta seção, serão abordados os principais termos, conceitos e ideias que serviram de suporte e incentivo para a realização deste trabalho e a construção de sua estrutura. Será explorado como esses fundamentos contribuíram para a formação da base teórica e prática do projeto.

2.1.1
Relacionamento entre pessoas

A espécie humana possui um forte caráter social, criando relacionamentos uns com os outros e sempre buscando se integrar em algum grupo. Estes relacionamentos influenciam não apenas indivíduos íntimos entre si, mas também os contextos nos quais estão inseridos. Estes últimos podem estar presentes em diversos ambientes, como o familiar, o laboral e o político, além de, em certas ocasiões, formarem interseções, isto é, combinações entre diferentes tipos de vínculo. No entanto, essas interações podem, por vezes, ser inadequadas para determinados contextos, gerando conflitos de interesse.

No ambiente de trabalho, a cooperação entre colegas é essencial. Como destaca Sergio Leitão ao abordar a estratégia de empresas do projeto de economia de comunhão (EdC), "(...) para haver aumento espontâneo da produtividade do trabalho, é preciso elevar o nível de qualidade nos relacionamentos entre todos que operam a empresa"(LEITãO; FORTUNATO; FREITAS, 2006).

O relacionamento entre indivíduos e o *networking* podem ser fatores que impulsionam tanto a coletividade quanto o desempenho profissional, favorecendo empresas e negócios. No entanto, essas mesmas relações podem se tornar instrumentos de interesse oculto, cruzando os limites estabelecidos pelas leis nacionais. Embora possam desenvolver negócios, essas práticas comprometem a integridade humana, tanto individual quanto socialmente.

2.1.2
Pessoas Expostas Politicamente (PEP)

As diversas definições de PEPs que podemos relacionar com dois termos principais: "influência- tanto política quanto social, o que amplia ainda mais seu conceito - e "proximidade", ou seja, todos que tenham um relacionamento próximo com o agente PEP em questão. Com relação a este último, é interessante apontar que, muitas vezes, essa relação de proximidade está fortemente associada a laços familiares, laços de amizade e laços de sociedade econômica,

como contratos, empresas, etc.

A principal definição encontrada que aborda sobre "proximidade"foi a do Banco Central do Brasil, que consta que o grupo PEP é formado por agentes públicos que desempenham ou tenham desempenhado, nos últimos cinco anos, cargos, empregos ou funções públicas relevantes (BRASIL, 2009). Seus representantes, familiares e outras pessoas de seu relacionamento próximo também estão incluídos nesta definição.

Outra definição, agora abordando apenas sobre "influência", é vinda da plataforma Gov.Br, que define uma pessoa como PEP os ocupantes de cargos e funções públicas listadas nas normas da Prevenção e Combate à Lavagem de Dinheiro e ao Financiamento do Terrorismo e da Proliferação de Armas de Destruição em Massa (PLD/FTP) (GOV.BR, 2025b).

2.1.3 Nepotismo

Baseado nas seções 2.1.1 e 2.1.2, é possível relacionar influência de pessoas próximas a PEPs com relações interpessoais de má índole. Nesta área social, fatores como manipulação e ganância podem resultar em dinâmicas sociais e de trabalho preocupantes. Uma destas dinâmicas é o nepotismo.

O nepotismo, de acordo com a plataforma oficial Gov.Br, ocorre quando um agente público usa de sua posição de poder para nomear, contratar ou favorecer um ou mais parentes (GOV.BR, 2025a). Vale mencionar também que, disponibilizado novamente pelo próprio Gov.Br, há diferentes tipos de nepotismo como nepotismo direto e cruzado:

> "Nepotismo direto é aquele em que a autoridade nomeia seu próprio parente. Nepotismo cruzado é aquele em que o agente público nomeia pessoa ligada a outro agente público, enquanto a segunda autoridade nomeia uma pessoa ligada por vínculos de parentescos ao primeiro agente, como troca de favores (...)."(GOV.BR, 2025a)

Pode-se concluir que o nepotismo se baseia totalmente em relacionamentos que trazem vantagem em algum meio político, econômico ou empresarial. Influência familiar dentro de contextos como estes valorizam a opinião e interesses da família em questão.

2.1.4
Corrupção e fraude

O sistema de Prevenção e Combate à Lavagem de Dinheiro e ao Financiamento do Terrorismo e da Proliferação de Armas de Destruição em Massa (PLD/FTP) é um conjunto de normas e procedimentos para combater a lavagem de dinheiro e o financiamento ao terrorismo. Dentre os órgãos regulamentadores e fiscalizadores do Sistema Brasileiro de PLD/FTP está o Conselho de Controle de Atividades Financeiras (Coaf), o qual é responsável pelo monitoramento de fomentos comerciais. Estes últimos se dão pela estratégia ou ação voltada para estimular e promover o desenvolvimento de atividades comerciais e empresariais, o que, consequentemente, envolve incentivos financeiros, técnicos ou estratégicos oferecidos por governos.

Pode-se perceber que a relação entre fomento comercial e PLD/FTP está na necessidade de garantir que as operações comerciais financiadas não sejam usadas para atividades criminosas. Logo, é possível entender o porquê dos termos "corrupção"e "fraude"estarem sendo abordados, não só com base no exemplo mostrado, mas também pelos termos citados anteriormente e os diversos outros órgãos fiscalizadores do sistema nacional de PLD/FTP.

2.2
Casos de problema

Nesta seção, iremos abordar casos de problemas que envolvem PEPs e que deram motivação para a realização deste trabalho.

2.2.1
Emendas parlamentares no Brasil

A busca da relação entre o planejamento e orçamento é o dever da administração pública e de seus atores (CARNUT et al., 2021). Por conta da grande diversidade e dimensão do território nacional, é um desafio elaborar um planejamento orçamentário integrado (CARNUT et al., 2021), já que diferentes regiões e culturas do país requerem demandas específicas e um custo específico.

As emendas parlamentares são regulamentações sobre o orçamento nacional que ditam quem, quando, quanto e onde os recursos serão destinados. Entre diversos tipos de emendas, existem as emendas do tipo PIX, as quais introduziram maior celeridade ao processo, dispensando a celebração de convênios e transferindo os recursos diretamente aos entes federados beneficiários (BARBOSA, 2024). Ou seja, as emendas PIX fogem de qualquer tipo de fiscalização e não possuem nenhuma transparência em seus registros, dando grande liberdade aos autores públicos sobre o orçamento nacional.

Anteriormente, em 2015, houve uma mudança drástica com relação às emendas parlamentares quando as emendas impositivas surgiram no cenário financeiro. Estas deram acessibilidade para os desvios de verba à escolha de cada senador, independente da opinião ou validação do presidente. Antes dessa mudança, as emendas eram sugestivas, ou seja, eram atendidas e aprovadas a partir de decisões de ministérios ou a partir do interesse do presidente.

Portanto, com a imposição dos desvios de verba que, posteriormente com as emendas PIX, se tornaram impossíveis de rastrear, as emendas parlamentares no Brasil se tornaram um meio invisível de corrupção e fraude entre políticos.

2.2.2
Favorecimento de empresas

Anteriormente foi abordado como interesses pessoais influenciam e inclinam decisões para a escolha de pessoas para cargos ou indicações. O mesmo

pode ocorrer em relação a empresas, especialmente em processos de licitação. Nesse contexto, o favorecimento acontece quando há comportamentos maliciosos que direcionam indevidamente a escolha para uma empresa específica.

Esta infração é grave, já que viola os princípios da isonomia (PúBLICO, 2015), da impessoalidade e da igualdade, constados na Lei de Licitações e Contratos Administrativos (JURíDICOS, 2021). Este princípio não admite qualquer tratamento diferenciado e apoia igualdade a todos perante à lei.

2.3 Ontologias

Diferente da organização de pesquisa antiga da Web, as ontologias procuram uma nova forma de atribuir sentido e contexto a uma busca. Ontologias são ferramentas ou linguagens que permitem a instauração de sentido, diminuindo a polissemia de um mesmo termo para assuntos diversos (PICKLER, 2007), ou seja, são ferramentas de contextualização para um domínio ao qual estão relacionados os termos em si.

As ontologias são bastante utilizadas no meio da computação e informática. Não só no contexto da Web, essas ferramentas também são consideradas na estrutura de banco de dados, delimitando e explicitando o domínio de uma base de conhecimentos oriunda do banco de dados propriamente dito. Um exemplo é a *University Ontology* (ONTOLOGIES, 2000), a qual abstrai muitos tipos de entidades em um ambiente acadêmico, como professores, estudantes, departamentos, diretores, universidades, entre outros. Esta estratégia possibilita que o público que não tenha acesso direto aos dados ou conhecimentos prévios de estrutura de dados entendam e façam uso do conhecimento ali exposto.

A estrutura de uma ontologia é composta de um vocabulário de termos utilizados em um domínio e uma especificação semântica deste vocabulário (MOREIRA, 2010). A forma mais comum na computação no uso de ontologias se baseia em descrever os seguintes termos (MORAIS; AMBRóSIO, 2007):

- Classes: representam algum tipo de interação da ontologia com um determinado domínio;
- Relacionamentos: tipo de interação entre os elementos do domínio;
- Axiomas: modelam sentenças consideradas sempre verdadeiras;
- Instâncias: representam elementos específicos, isto é, os próprios dados da ontologia;
- Funções: eventos que podem ocorrer no contexto da ontologia.

3
Objetivos

Os objetivos principais deste trabalho são a definição, a construção e a disponibilização de uma base de conhecimentos sobre PEPs e seus respectivos relacionamentos de âmbitos familiares e comerciais. A base de conhecimento - ou *Knowledge Base* (KB)- permitirá a realização de consultas em linguagem natural.

Para a definição das regras, limitações e escopo da base de conhecimento, fizemos uso de mais de uma ontologias como apoio por conta do tema, contexto e tipos de dados que foram utilizados.

A alimentação desta KB será tanto automática quanto manual e permitirá a existência de diversas fontes com diferentes graus de confiabilidade, os quais serão definidos durante o processo. Como exemplo de fontes de informação consideradas são as plataformas da Wikipédia, Tribunal Superior Eleitoral (TSE) e indivíduos com certo grau de confiança, por exemplo jornalistas.

A consulta na base se dará por meio de uma *Application Programming Interface* (API) ou de uma interface *Web*, e a consulta em si é no formato de linguagem natural. Posteriormente, a resposta é processada e preparada por um *Large Language Model* (LLM) a partir do conteúdo selecionado na base de conhecimento, o que caracteriza um *Retrieval Augmented Generation* (RAG). A resposta também é dada na forma de um grafo no formato JSON.

4
Metodologia

O diagrama do metodologia de construção do *software* proposto se encontra na Figura 4.1.

Na metodologia, separamos processos de construção - desde definições de conceitos até implementação - para cada componente do sistema. Para a alimentação do banco de dados, definimos os requisitos para a captura e consulta de dados, as regras de governança de dados e o tratamento de dados em si para popular o banco.

A base de conhecimento será construída através da definição e pesquisa de ontologias a serem usadas, da conceituação de sua modelagem, da pesquisa sobre modelos compatíveis de LLMs, da alimentação do banco em si e da construção de um coletor de dados (*retriever*).

Por último, o componente de consulta ao banco será construído através de três funcionalidades distintas: a integração caracterizada como RAG, a construção da interface *Web* e a construção de uma API.

Por fim, temos testes de validação e finalização do sistema proposto com base em questões de competência propostas, as quais falaremos mais à frente.

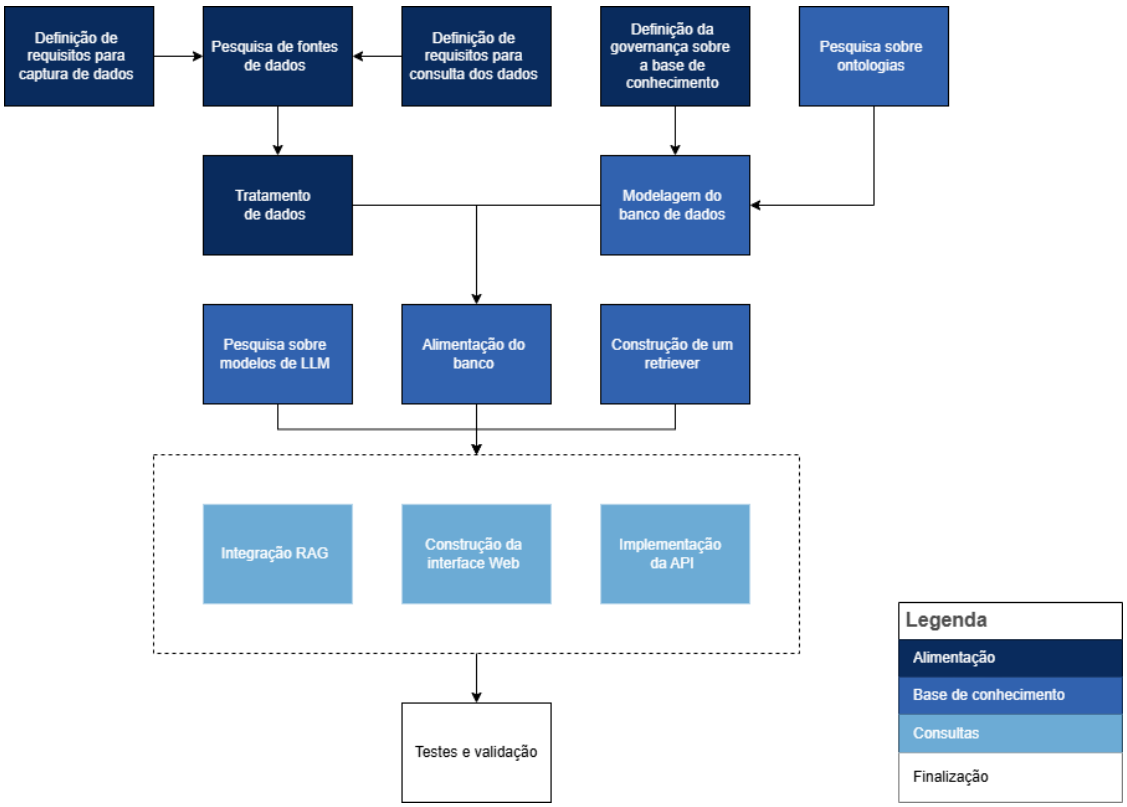

Figura 4.1: Diagrama do processo de construção do software

5
Estudos

Estudos e pesquisas foram realizados para entender quais ferramentas, tecnologias, conceitos e definições seriam interessantes para este trabalho.

5.1
Base de conhecimento

Para a base de conhecimento, foram buscadas definições que fizessem mais sentido para este trabalho. Foram encontradas definições que constam armazenamento de dados não-estruturados (LIVRE, 2023) e que abordam sobre a base de conhecimento ser uma biblioteca online sobre um produto ou serviço, e que possíveis contribuidores integrados no domínio podem agregar para a base (ATLASSIAN, 2025).

Técnicas e ferramentas de como criar uma modelagem conceitual orientada à grafo foram pesquisadas, desde o básico deste tipo de modelagem (LINKURIOUS, 2025), conversão de um banco de dados relacional (DISNEY, 2020) até um novo conceito sobre bases de conhecimento temporais/sensíveis ao tempo chamados *Temporal Knowledge Graphs* (TKG) (DOGAN, 2024) (KNEZ; ŽITNIK, 2023).

A governança sobre os dados é um fator indispensável neste projeto. Com as pesquisas e estudos feitos, foram descobertos que não há uma definição clara do que seria governança de dados, mas que é um assunto bastante falado no campo da computação - abordando principalmente sobre *IT Governance* - e que é um fator que leva em consideração processos organizacionais (NIELSEN, 2017). Também foram descobertos alguns princípios fundamentais de governança de dados, como (BROUS; JANSSEN; VILMINKO-HEIKKINEN, 2016):

- Organização: definição da estrutura de governança, como direitos de decisão e responsabilidades;

- Alinhamento: definição de políticas e estratégias de qualidade de dados que se baseiem nas prioridades estratégicas da organização;
- Conformidade: práticas de segurança, privacidade e responsabilidade;
- Entendimento Comum: modelos de dados e padronização de metadados para que os dados sejam bem compreendidos.

O uso de ontologias também é relevante. Foi encontrado um estudo de construção de ontologias a partir da estrutura semi-estruturada da plataforma *Wikipédia*, em que se concluiu que, por conta de sua estrutura, a plataforma é rica e viável (RAJEBAH; AL-KHALIFA, 2010).

5.2 Alimentação do banco

Estudos sobre métodos e técnicas para a alimentação do banco de dados baseou-se tanto em possíveis *datasets* para extração de dados quanto em estudos de suas respectivas APIs e que tipo de dados elas fornecem.

Para possíveis APIs de fontes confiáveis com dados igualmente confiáveis, temos a API Dados Abertos da Câmara dos Deputados (DEPUTADOS, 2025) e a API de Dados do Portal da Transparência da plataforma GOV.Br (UNIãO, 2025). Especificamente, para popular informações sobre relações familiares, temos a API da Wikimedia (Wikimedia Foundation, 2025). Para relacionamentos dentro do âmbito comercial e empresarial, temos a API do Tribunal Superior Eleitoral (ELEITORAL, 2025).

Para entender melhor o que precisaria ser inserido no *pipeline* de tratamento e padronização de dados, foi encontrada a biblioteca APOC (TEAM, 2025), disponibilizada pelo Neo4j. Esta biblioteca possibilita e facilita que tipos diferentes de arquivos - como CSV e JSON - possam ser interpretados e carregados para a base de dados.

Para cada fonte de dados, serão atrelados graus de confiabilidade que representam o quão verossímil e fiel é aquela informação. Pela pesquisa, destacou-se a importância da definição da proveniência dos dados, ou seja,

de todo o histórico de como aquele dado foi gerado, modificado e manipulado (ZAFAR et al., 2017) para que ele seja considerado confiável. Também há a verificação da *provenance chain*, ou seja, da cadeia de custódia dos dados que pode revelar potenciais mudanças maliciosas ao longo do histórico dos dados (ZAFAR et al., 2017).

As personas definidas que servirão como contribuidoras da base de conhecimento são pessoas de cargos da imprensa, como jornalistas, pessoas inseridas no meio acadêmico, como professores da área política, e pesquisadores de dados que pertencem ao domínio de conhecimento da base, como analista de dados.

5.3
Consultas ao banco

É importante lembrar que as consultas do sistema se darão por dois meios diferentes: interface Web com apoio de um modelo de LLM e API com respostas com sub-grafos em arquivos JSON.

Foram encontrados não apenas um modelo de LLM específico de tratamento de linguagem natural (NLP) para consultas em Cypher - linguagem oficial usada para bases disponibilizadas pelo Neo4j - (NEO4J, 2025), mas também como criar uma integração com diversas bibliotecas - aqui referenciadas LangChain e Neo4j GraphRAG - que criem uma dinâmica do tipo *Retrieval Augmented Generation* (RAG) (GAUTAM, 2024).

Para a construção da API, foi encontrado o framework *low-code Postman Flows* (POSTMAN, 2025), que possibilita uma construção e manutenabilidade de API sem maiores complexidades.

6
Requisitos

Nesta seção, os requisitos, condições e obrigatoriedades de cada componente do sistema é definido, com base no objetivo principal do sistema e nos estudos que se deram neste relatório.

6.1
Base de conhecimento

Para a *knowledge base* (KB), os seguintes requisitos são definidos:

6.1.1
Requisitos Funcionais

1. A KB deve se basear em ontologias;
2. A KB deve considerar graus de confiabilidade para nós e arestas;
3. A KB deve considerar temporalidade para arestas;
4. A KB deve considerar a extensibilidade para arestas para que novos relacionamentos possam ser agregados;
5. A KB deve ter restrições para operações de acesso e manipulação;
6. A KB deve conter metadados;
7. A KB deve ser capaz de responder questões de competência.

6.1.2
Requisitos Não Funcionais

1. A KB deve ter uma modelagem conceitual;
2. A KB deve ser orientada à grafo, por conta da melhor representação de relacionamentos entre entidades;
3. A KB deve ter backup completo e rotinas de backup.

6.2
Alimentação do banco

Para os métodos e regulamentações de alimentação do banco, temos os seguintes requisitos:

6.2.1
Requisitos Funcionais

1. O sistema deve recolher dados confiáveis de diversas fontes igualmente confiáveis;
2. O sistema deve realizar um pipeline de tratamento e padronização dos dados anterior à alimentação;
3. O sistema deve ter rotina(s) automatizada(s) de alimentação;
4. O sistema deve disponibilizar uma opção manual de alimentação;
5. O sistema deve registrar a proveniência dos dados;
6. O sistema deve autenticar as fontes de dados automáticas e manuais;

6.2.2
Requisitos Não Funcionais

1. O sistema deve ter controle sobre inserções simultâneas no banco;
2. O sistema deve ter controle de acesso por perfil de usuário.

6.3
Consultas ao banco

Para os métodos que possibilitam consultas ao banco de dados, temos os seguintes requisitos:

6.3.1
Requisitos Funcionais

1. As consultas devem poder ser feitas através de uma API e/ou uma interface Web;
2. A API deve ter um formato GET e POST pré-definido para formato JSON que represente um sub-grafo como resposta;
3. As consultas devem ser de acesso e uso públicos;
4. O sistema deve conseguir tratar eventuais erros na consulta;
5. O sistema deve conseguir tratar eventuais resultados nulos;
6. As consultas na interface Web devem ser respondidas por um modelo de LLM integrado em uma técnica de *Retrieval-Augmented Generation* (RAG), ou seja, devem ser respondidas em linguagem natural;
7. O resultado da consulta, caso seja dado como sub-grafo em JSON, deve conter metadados.

6.3.2
Requisitos Não Funcionais

1. A API deve ter um formato JSON pré-definido.

7
Especificação

Nesta seção, é abordado especificações e estruturas de cada componente e sub-componente deste sistema. As especificações aqui descritas foram elaboradas com base nos requisitos definidos, garantindo o alinhamento entre os objetivos do sistema e as soluções propostas. Todos os requisitos estabelecidos foram considerados e integralmente cobertos pelas funcionalidades especificadas, em especial no que se refere à *Knowledge Base*.

7.1
Modelagem, Ontologias & Questões de Competência

Para a elaboração da modelagem do banco, foram pesquisadas diversas ontologias que fossem interessantes para o domínio de informação do banco de dados. A pesquisa levou em consideração possíveis entidades - representadas pelos nós no grafo -, como Parceria e Organização, e diversos relacionamentos - representados pelas arestas entre os nós no grafo - que podem ser representados por uma ou mais ontologias. Relacionamentos, parentescos, ocupação de um cargo em uma empresa, realização de acordos, entre outros, também são exemplos do tipo de domínio de informação que queremos cobrir.

Na modelagem, mostrada na Figura 7.1, é possível perceber que os nomes das entidades, dos relacionamentos e respectivos valores dos atributos possuem prefixos, cada qual faz menção a uma ontologia, e sufixos, que referenciam classes ou propriedades de classes da ontologia em prefixo. Por exemplo, o prefixo "fibo:"aponta para a *Financial Industry Business Ontology* (FIBO). O sufixo "Partnership"é uma classe de uma ontologia que segue o domínio e especificação da FIBO.

Intrinsecamente, a modelagem a ser implementada não utilizará diretamente prefixos e *labels* de classes e atributos oriundos das ontologias citadas aqui. As ontologias são utilizadas na modelagem principalmente como inspi-

ração, como dicionário de dados e como entendimento dos diferentes padrões e modelos de formatação de conhecimento. Não usaremos as ontologias para nenhum tipo de automação e não definiremos nossa própria ontologia.

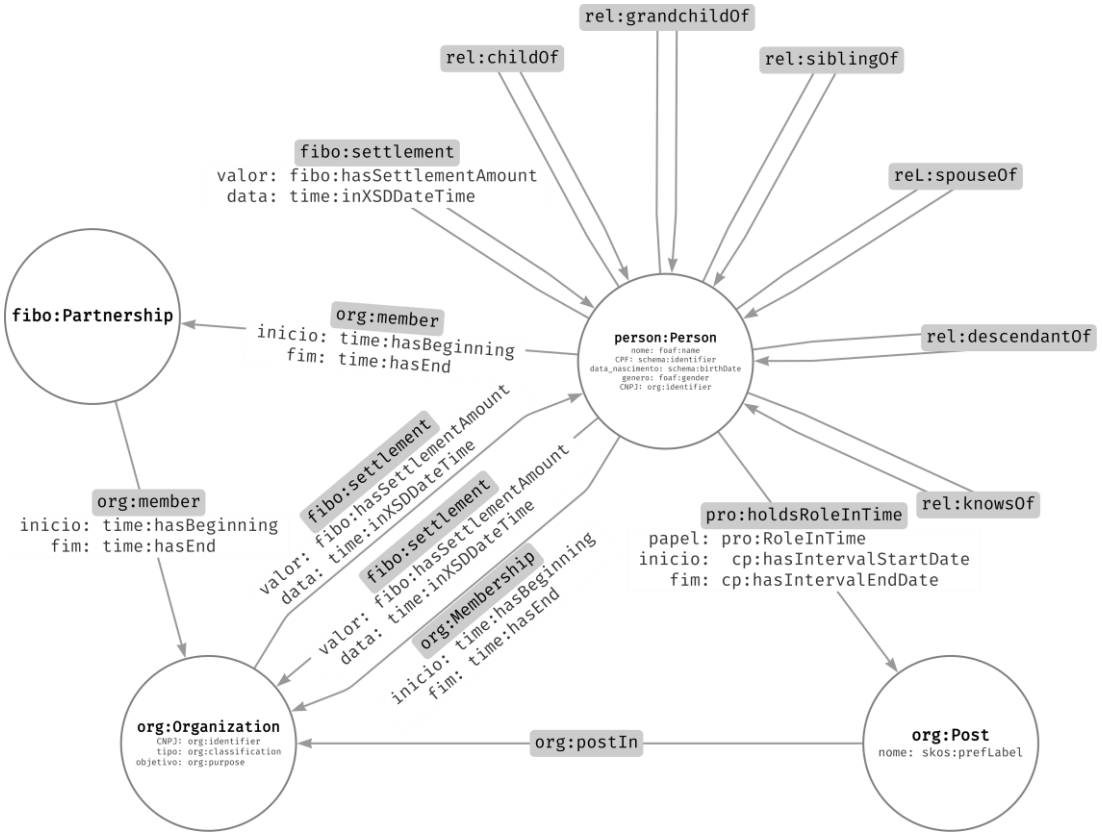

Figura 7.1: Modelagem conceitual orientada à grafo e à ontologias.

Esclarecendo melhor cada componente e definição, listamos aqui os prefixos, suas respectivas ontologias e padrões - os quais usam uma ou mais ontologias para seus domínios - e seus atributos usados na modelagem:

- **foaf:** : ontologia *Friend Of A Friend* (FOAF), usada para ligar pessoas e informações pela Web.
  - **foaf:name** : atributo que representa o nome de alguém ou de alguma coisa;
  - **foaf:gender** : representa o gênero de uma pessoa.
- **person:** : referencia o padrão *Core Person Vocabulary* (CPV), ontologia que procura padronizar características fundamentais de uma pessoa, como data de nascimento;

  - **person:Person** : classe que representa a entidade de uma pessoa, física ou jurídica.

- **fibo:** : aponta para a ontologia *Financial Industry Business Ontology* (FIBO), que define classes e atributos que sejam interessantes em aplicações comerciais e financeiras;

  - **fibo:settlement** : representa uma finalização de uma transação financeira, registro de documentação ou trocas de considerações;
  - **fibo:hasSettlementAmount** : o valor monetário requerido por uma transação financeira.

- **rel:** : prefixo que representa a ontologia *Relationship*, a qual descreve relacionamentos entre pessoas;

  - **rel:childOf** : relacionamento de uma pessoa que nasceu ou foi criada por outra pessoa;
  - **rel:grandchildOf** : representa a relação de uma pessoa que é neta/neto de outra pessoa;
  - **rel:siblingOf** : representa a relação de uma pessoa que é irmã/irmão de outra pessoa;
  - **rel:spouseOf** : representa a relação de uma pessoa que é cônjuge de outra pessoa;
  - **rel:descendantOf** : representa a relação de uma pessoa que é qualquer outro tipo de descendente de outra pessoa;
  - **rel:knowsOf** : representa a relação de uma pessoa que conhece outra pessoa.

- **time:** : referencia a ontologia *Time Ontology*, a qual busca descrever propriedades temporais dos recursos do mundo e da Web;

  - **time:hasBeginning** : associa um intervalo de tempo a um momento em que ele começa;

  - **time:hasEnd** : associa um intervalo de tempo a um momento em que ele termina;
  - **time:inXSDDateTime** : propriedade de dado que define um momento temporal específico.

- **pro:** : aponta para a *Publishing Roles Ontology* (PRO), que caracteriza papéis de agentes como pessoas, corporações, entre outros;
  - **pro:RoleInTime** : classe que representa o papel desempenhado por um agente em um determinado período de tempo;
  - **pro:holdsRoleInTime** : propriedade que associa um agente ao papel que ele exerceu durante um determinado intervalo temporal.

- **org:** : representa a *Organization Ontology*, que descreve uma ontologia para estruturas organizacionais;
  - **org:Post** : classe que representa um cargo ou posição dentro de uma organização;
  - **org:Organization** : classe que representa uma organização, instituição ou empresa;
  - **org:Membership** : classe que representa a associação de uma pessoa ou agente a uma organização;
  - **org:postIn** : propriedade que liga um cargo (*Post*) à organização em que ele está inserido;
  - **org:identifier** : representa um identificador único de uma organização, como CNPJ ou outro código;
  - **org:purpose** : descreve o propósito ou a finalidade de uma organização;
  - **org:classification** : classifica a organização segundo um tipo ou categoria (por exemplo, pública, privada, ONG etc.);
  - **org:member** : associa um agente ou pessoa como membro de uma organização.

- **skos:** : prefixo que aponta para a *Simple Knowledge Organization System Reference* (SKOS), padrão de modelo de dados para compartilhamento de sistemas de conhecimento organizacional pela Web.
  - **skos:Concept** : representa um conceito dentro de um sistema de conhecimento;
  - **skos:prefLabel** : rótulo preferencial usado para identificar um conceito de forma legível;
  - **skos:broader** : indica uma relação hierárquica em que um conceito é mais geral que outro.
- **cp:** : a *Corporate Bodies, Persons and Families Relationships Ontology* (CPF-Relationships) é representada por este prefixo e descreve relacionamentos entre corporações, pessoas e famílias;
  - **cp:hasIntervalStartDate** : indica a data de início de um intervalo temporal relacionado a uma função, relação ou evento;
  - **cp:hasIntervalEndDate** : indica a data de término de um intervalo temporal relacionado a uma função, relação ou evento.
- **schema:** : a *Schema.org* é um vocabulário que abrange entidades, relacionamentos entre entidades e ações e pode ser facilmente estendido.
  - **schema:identifier** : o atributo "identifier"representa qualquer tipo de identificador, como números ou códigos únicos;
  - **schema:birthDate** : representa a data de nascimento de uma pessoa, geralmente expressa no formato AAAA-MM-DD;
  - **schema:affiliation** : descreve a afiliação de uma pessoa a uma organização;
  - **schema:description** : fornece uma descrição textual de uma entidade.

7.1.1
Questões de competência

Na área de desenvolvimento de ontologias, para a etapa de validação de uma ontologia formulam-se questões de competência (*Ontology Competency Questions*), as quais determinam a expressividade de uma ontologia, ou seja, são perguntas as quais uma ontologia é capaz de responder (NOY; HAFNER, 1997). Aqui, as questões de competência são perguntas em que a modelagem baseada em ontologias consegue responder, e, consequentemente, perguntas em que o sistema consegue responder.

As questões de competência também são usadas por muitos como um tipo de validação no desenvolvimento de uma ontologia (BEZERRA; FREITAS; SANTANA, 2013). Dentre as etapas mais relevantes e utilizadas na construção de ontologias, a definição do escopo é feito principalmente por questões de competência.

Com isso, aqui estão as questões de competência a serem verificadas pelo sistema:

- "Onde a pessoa X trabalha?";
- "Que pessoas trabalham em X?";
- "Quem são os irmãos da pessoa X?";
- "Qual é a ligação/relacionamento entre as pessoas X e Y?";
- "Quais são os parentes da pessoa X?";
- "Quais são os irmãos da pessoa X que trabalham na empresa Y?"

Essas questões servirão não apenas como base das funcionalidades principais do sistema, mas também como base de casos de teste para a etapa de validação.

7.2
Manutenção & Governança de Dados

A governança de dados é indispensável neste cenário. Ditamos como governança de dados o planejamento e controle de alto nível sobre o gerenciamento de dados (AL-RUITHE; BENKHELIFA; HAMEED, 2019). O sistema possibilita que certos tipos de usuários consigam contribuir para a base de conhecimento adicionando informações, como novas entidades ou novos relacionamentos no grafo. Por conta disso, métricas de governança de dados são importantes para que sejam categorizados rigorosamente os tipos de usuários contribuidores, manter a consistência da modelagem do banco, manter a consistência dos dados sem repetições ou conflitos, entre outros.

Por conta dos diversos potenciais usuários contribuidores, o banco de dados necessita de estratégias e métricas para lidar com diversos tipos de incertezas e riscos, levando em consideração os tipos de informações novas e os diferentes graus de confiabilidade atribuídos aos contribuidores e suas respectivas contribuições.

O banco de dados irá persistir dados com diferentes graus de confiabilidade, definidos em relação ao domínio de informação do projeto. Esse grau determina o quão confiável uma informação é considerada dentro desse domínio específico e decorre da diversidade de fontes de dados disponíveis e passíveis de utilização. O grau de confiabilidade adota uma escala de 5 pontos para a classificação das fontes, na qual o valor 5 representa uma fonte altamente confiável no contexto do domínio do projeto, enquanto o valor 1 representa uma fonte considerada pouquíssimo confiável.

Adicionalmente, conceitos de reputação discutidos na literatura de sistemas de confiança e reputação comprovam a ideia de graus de confiabilidade adotados neste trabalho. Nessas abordagens, reputação é entendida como a avaliação agregada de quão confiável ou bem considerada uma entidade é em um determinado contexto, derivada de experiências ou opiniões de múltiplos avaliadores (JØSANG; ISMAIL; BOYD, 2007). Isso se alinha com a definição

de grau de confiabilidade aqui utilizada, em que cada fonte de informação recebe um valor em uma escala para indicar sua confiança relativa dentro do domínio do projeto.

A pesquisa sobre atribuição de graus de confiança (muitas vezes chamados de *data reliability score*) apresentou resultados interessantes porém com alta complexidade para este trabalho. Graus de precisão de dados, incluindo para grafos de conhecimento, são criados e atribuídos por algoritmos complexos ou por modelos de *Machine Learning*, o que foge do escopo deste projeto. Assim, cada informação nova persistida no banco de dados herdará o mesmo grau de confiança de sua respectiva fonte.

Algumas das estratégias consideradas para governança de dados resumem-se a seguir:

1. O gestor do sistema será o agente atribuinte de graus de confiabilidade de cada fonte de dados;
2. Toda informação persistente no banco - sendo esta arestas ou nós no grafo - mostrará de alguma forma o seu grau de confiabilidade;
3. Para inserção de diferentes instâncias sobre uma mesma informação com diferentes graus, apenas a instância de maior grau de confiabilidade permanecerá;
4. Para a estratégia anterior, em caso de graus iguais, ambas as informações serão persistidas;
5. Os usuários contribuintes terão restrições para deleção de dados persistidos, podendo apenas remover informações de suas próprias colaborações;

Com essas estratégias, estamos abrangendo soluções tanto para fatos errôneos quanto para fatos ambíguos, como abordam Yang *et al.* (YANG; CHEN; XIANG, 2025). Ou seja, dando prioridade à informações com maiores graus de confiabilidade, estamos aumentando o grau de confiança na nossa

base de conhecimento e evitando conflitos e repetições entre os dados. Com a implementação da estratégia número 4, por exemplo, estamos abordando uma solução para fatos ambíguos. Isso significa que, como estamos englobando diversas fontes de dados, é possível ter diversas perspectivas sobre uma mesma informação. Assim, incluindo ambas as informações de bons graus de confiança, englobamos todas as perspectivas de um pedaço de conhecimento.

8
Arquitetura

A arquitetura do sistema se concentra em três estruturas principais: ETL, Base de Conhecimento, interface Web e RAG.

A estrutura Extract, Transform & Load representa quatro operações fundamentais para manipulação de dados em software. Os *scripts* ETL recolhem os seguintes tipos de dados das seguintes fontes de dados:

1. Plataforma Wikipédia para recolhimento de dados de familiares de PEPs;
2. Documento CSV com dados sobre mandatos de PEPs. Esse grande documento foi retirado manualmente da plataforma do Portal de Dados Abertos Gov.Br (UNIãO, 2025);
3. Dados inseridos sobre sociedades, cotas e bens de PEPs. Estes dados foram retirados do Portal de Dados Abertos Gov.Br (UNIãO, 2025), do conjunto de dados do TSE (ELEITORAL, 2025) e da plataforma Redesim (REDESIM, 2018).

A Base de Conhecimento é a segunda estrutura deste sistema, a qual as especificações e definições já foram mencionadas.

A interface Web se resume em uma simples página HTML em que o usuário pode realizar perguntas em uma caixa de busca. Por conta do grande escopo deste projeto, decidimos substituir a API em uma *feature* na própria interface Web, onde uma *checkbox* pode ser marcada para que um texto em formato JSON apareça como resposta final.

Sobre o componente RAG, sua arquitetura e fluxo está explicado da seguinte maneira pela figura 8.1:

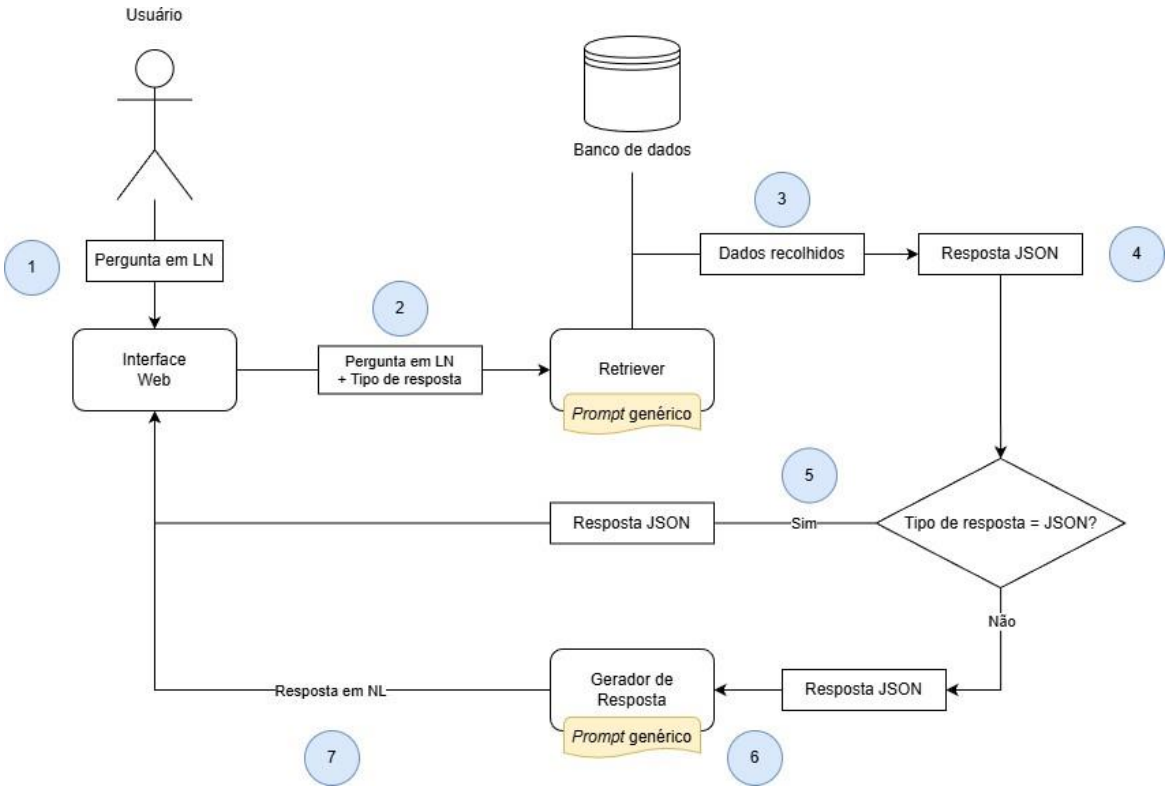

Figura 8.1: Fluxo de pergunta e resposta do RAG.

1. O usuário realiza uma pergunta em Linguagem Natural (LN) pela interface Web. Ao marcar ou não a *checkbox* na interface, ele também escolherá o tipo de resposta que deseja;

2. A pergunta e o tipo de resposta são repassados para o *retriever* do sistema;

3. O *retriever* faz o recolhimento dos dados. Para isso, constrói consultas e as aplica no banco de dados;

4. Com os dados recolhidos, o fluxo RAG constrói uma resposta em formato JSON;

5. Caso o tipo de resposta escolhido pelo usuário for em formato JSON, o JSON formado é repassado diretamente para a interface;

6. Caso o tipo de resposta for em LN, o JSON é repassado para o componente Gerador de Resposta que gera uma resposta final em LN;

7. Por fim, a resposta em LN é repassada para a interface Web.

Vale ressaltar também que, como representado na figura 8.1, tanto o componente *Retriever* quanto o componente Gerador de Resposta possuem um ***prompt* genérico**, representado em amarelo. Cada componente possui seu próprio *prompt*, cujo objetivo é fornecer ao modelo o **contexto**. O contexto fornecido aos LLMs atua como um mecanismo de *input conditioning*, no qual informações adicionais são utilizadas para orientar e restringir a geração do modelo (LIU et al., 2023).

No *retriever*, seu *prompt* recebe 1. a pergunta do usuário, 2. a estrutura do banco de dados e 3. exemplos de consultas ao banco. No componente de Gerador de Resposta para a resposta em LN, seu *prompt* recebe 1. o *prompt* do *retriever* e 2. o JSON formado.

Ou seja, isso significa que tanto o *Retriever* quanto o Gerador de Resposta possuem suas próprias instâncias de LLM para geração de consultas e geração de resposta em LN, respectivamente. Cada *prompt* nada mais é do que um *input* para cada LLM ser capaz de produzir o que se pede.

Os respectivos *prompts* genéricos de cada um desses componentes está descrito abaixo:

1. *Retriever* :

> "Você é um agente especialista em recolher dados sobre cargos, familiares de políticos brasileiros e organizações brasileiras. Dada uma pergunta do usuário, você formará uma query Cypher que responda a pergunta.
>
> O schema do banco de dados é este:
>
> {context}
>
> LEMBRE DISSO: Uma Organizacao pode representar tanto um órgão do governo quanto um estado ou município.
>
> Você pode retornar nós do banco ou caminhos entre nós.

Sempre entregue relacionamentos de maior grau_precisao, se for pedido.

Aqui estão alguns exemplos:

{examples}

SEMPRE use UPPERCASE para os nomes dos nós!!

Use 'CONTAINS' para buscas parciais em strings.

Atente-se como são retornados caminhos entre nós: SEMPRE use [*1..], sem definir um comprimento fixo.

Pergunta do usuário:

{query_text}"

2. Gerador de Resposta:

"Dado o contexto:

'{PROMPT_TEMPLATE.format(query_text=pergunta)}'

e dado os dados de resultado:

{resultados_processados_list},

forme uma resposta final completa apenas sobre todas as informações dos dados recolhidos, não sobre seus conhecimentos gerais ou sobre a query feita."

9
Construção & Tecnologias

Para a construção do sistema, foram escolhidos os *frameworks django* para a interface Web e *neomodel* para a integração com o banco de dados Neo4j através do próprio *django*.

9.1
Modelagem final do banco

A modelagem baseada em grafo e ontologias foi traduzida em outra modelagem com a mesma topologia - que se refere a como os nós e arestas estão organizados e conectados - porém com títulos dos nós e dos relacionamentos mais intuitivos. Seguindo a sugestão da própria plataforma do Neo4j, os nomes dos relacionamentos contém verbos no presente, que identificam características ou ações entre nós. Especificamente sobre os atributos, seus respectivos nomes já tinham sido pensados na modelagem Neo4j e as ontologias foram usadas como os valores de cada atributo.

Banco de dados Neo4j podem ser modelados de maneiras mais livres comparadas com outras plataformas. Não é necessário criar um *schema* detalhado antes com entidades, tabelas ou tipos de dados para cada atributo: é possível construir um banco de qualidade apenas por meio da inserção de dados. A cada inserção, o *schema* é atualizado para acompanhar novos tipos de relacionamentos e nós. Mesmo com certa liberdade, uma modelagem prévia é de extrema importância para este projeto e seus objetivos.

A Figura 9.1 apresenta a modelagem Neo4j orientada à grafo realizada. Esta modelagem dependeu totalmente no mapeamento feito na modelagem anterior com ontologias. Além disso, esta nova modelagem segue padrões e sugestões da plataforma Neo4j, como já dito anteriormente e, agora, também inclui grau de precisão nos nós.

Esta modelagem também está previamente modelada em código Python

no repositório do projeto em classes do tipo Model. Os Models são usados em *django* como fontes de dados e podem representar tabelas em bancos relacionais, por exemplo. Aqui, os Models representam entidades no grafo e são definidos usando especificações do *neomodel*, além de também definirem os relacionamentos entre entidades.

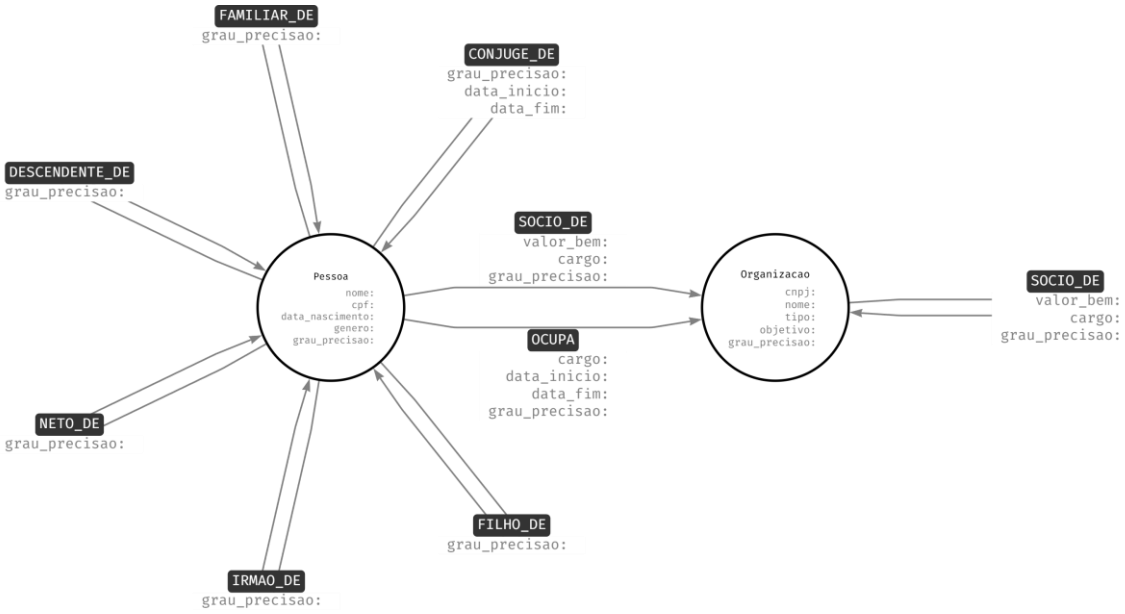

Figura 9.1: Modelagem conceitual orientada à grafo realizada.

9.2
Coleta e tratamento de dados

Como primeiro passo da construção desta prova de conceito de sistema, foram criados diversos *jupyter notebooks* para *scripts Extract, Transform and Load* (ETL), ou seja, algoritmos para que os dados sejam extraídos, transformados e carregados em um banco de dados. Felizmente, grande parte dos dados seguiam muitos padrões que facilitaram a limpeza e transformação dos dados.

Para a extração de dados na Wikipédia, foi construído um *script* com apoio do *parser BeautifulSoup* para páginas HTML - sendo estas as páginas bibliográficas de PEPs, caso existam - e, assim, um *match* para tipos de dados desejados eram recolhidos.

Dados sobre sociedades, valores de bens de PEPs e porcentagem de cotas encontrados no Portal de Dados Abertos Gov.br (UNIãO, 2025), no conjunto de dados do TSE (ELEITORAL, 2025) e na plataforma Redesim (REDESIM, 2018) foram recolhidos manualmente. Nomes de empresas de PEPs, sociedades e respectivos sócios foram procurados dentre essas três fontes, e novas entidades foram inseridas manualmente no banco.

9.2.1 Desafios encontrados

Os principais desafios desta seção se concentraram na coleta de dados da plataforma da Wikipédia. Por ser uma enciclopédia livre tanto para visualização quanto para edição pelo público, a alta inconsistência da estrutura HTML de uma página bibliográfica dificulta muito a coleta de dados. A falta de um padrão definido e a ampla variedade de nomenclaturas para os mesmos tipos de dados por toda a plataforma foi, sem dúvida, a maior dificuldade nesta parte do projeto.

Por conta da inconsistência dos dados bibliográficos, foi necessário adaptar todos os *scripts* de coleta e inserção para que, por exemplo, dado um nome de uma pessoa da Wikipédia verifica se ela já existe no banco de dados, mesmo que o nome da Wikipédia não esteja completo. Essas diferenças significativas entre as duas fontes de dados motivou muitas mudanças e adaptações de código.

A API por parte da plataforma é muito limitada, apenas disponibilizando pesquisas simples e impossibilitando coleta de dados mais específicos sobre uma página bibliográfica, como o nome de uma pessoa, seu nascimento, nome completo, etc.

9.2.2 Limitações

Uma das limitações da coleta de dados, além dos desafios citados, é o armazenamento do banco de dados. Com uma assinatura gratuita da instância

do banco Neo4j AuraDB, há uma limitação para a quantidade de dados armazenados. Por isso e por questões de demonstração do objetivo do sistema, foi inserida uma quantidade reduzida dos dados originais. Apenas os dados de PEPs abrangem mais de 130 mil instâncias de cargos diferentes e, caso todos tivessem sido inseridos, a maior parte do armazenamento do banco estaria preenchida.

9.3
Retrieval-Augmented Generation (RAG) & Interface Web

Para a implementação da dinâmica RAG, fizemos uso de uma ferramenta já conhecida: a classe Text2CypherRetriever, implementada pela plataforma Neo4j. Esta classe encapsula um *retriever* que, ao mesmo tempo que forma a consulta em Cypher a partir de uma pergunta em linguagem natural, também automaticamente recolhe dados do banco a partir da consulta formada.

Para este *retriever*, devemos considerar diversos parâmetros, como o esquema (*schema*) do banco de dados, uma instância de um LLM de sua escolha para geração da consulta, a instâncida de acesso ao próprio banco de dados (como um *driver*), exemplos de respostas esperadas e, por fim, um *prompt* customizado que descreva as funções da LLM. Vale ressaltar a facilidade de escolher e trocar o modelo LLM de raciocínio do *retriever*.

Depois da coleta feita pelo *retriever*, os dados resultantes e contextos iniciais são então passados para outra instância de LLM, a qual vai formular uma resposta em linguagem natural para o usuário final. Esta instância de LLM pode ser o mesmo modelo do *retriever* e também é facilmente mudável.

O modelo "command-r7b-12-2024"da plataforma Cohere foi usado para ambas instâncias de LLM e foi escolhido principalmente por sua falta de custo para uso.

A seguir, nas figuras 9.2, 9.3 e 9.4, são mostradas a interface Web realizada de acordo com as especificações de arquitetura e resultados em LN e em JSON:

Figura 9.2: Interface Web.

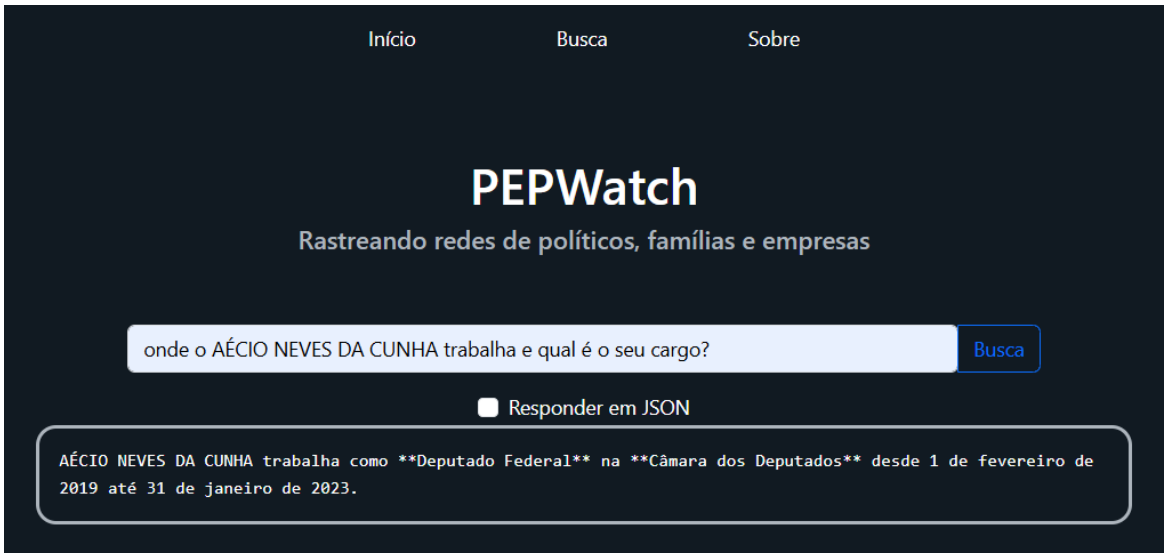

Figura 9.3: Exemplos de resultado em LN.

Figura 9.4: Exemplos de resultado em JSON.

Vale ressaltar que, dependendo da extensão da resposta final, independente do seu tipo, ela pode ser compactada em uma caixa com a opção de *scroll* para toda a sua visualização na interface.

9.3.1 Desafios encontrados

Um dos principais desafios encontrados foi a extração de todos os nós e relacionamentos de um **caminho** retornado pelo Text2CypherRetriever. Esse desafio se deu por conta do formato do retorno do retriever que não explicita todo o caminho em si. O retorno é encapsulado em uma classe própria do Neo4j e, no caso de um caminho retornado, as informações mais explícitas são o nó de início do caminho, nó final do caminho e o tamanho do caminho.

Para extrair informações de caminhos ou quaisquer informações retornadas, foi necessário criar funções auxiliares que desencapsulam as informações retornadas e as adicionam em dicionários que, posteriormente, são convertidos em um JSON.

9.3.2 Limitações

O modelo de LLM usado é disponível para uso da API sem custo algum. Em contrapartida, há limitações sobre sua capacidade de raciocínio e maior potencial de alucinações. Aqui, ambas as limitações do modelo influenciam drasticamente na resposta provida para o usuário.

Com esse tipo de limitação, também temos que, mesmo com o contexto dado e regras definidas para a geração da consulta, o modelo de linguagem por muitas vezes não segue as regras estabelecidas pela falta de capacidade de raciocínio, como já mencionado. Um exemplo é a regra de uso de nomes com todas as letras maiúsculas, onde, mesmo que o modelo consiga formar uma consulta coerente, nenhum resultado é encontrado sem que a regra seja seguida e, por fim, a resposta de erro "Não foi possível encontrar resultados para a pergunta fornecida. Tente reformular a pergunta."é retornada. Este cenário está mostrado na figura 9.5:

Figura 9.5: Resultado abrangendo limitações da LLM.

Com isso, adotei a seguinte estratégia: depois de um certo número de tentativas, a resposta de erro é novamente retornada. Esse espaço de tentativas possibilita um acerto maior da LLM ao gerar a consulta, sem interromper o fluxo do código.

Este último problema tem relação com uma última limitação: o modelo de LLM do *retriever* se baseia bastante nos exemplos de consulta presentes no seu *prompt* genérico. Isto pode acarretar na restrição do tipo de pergunta que o usuário pode fazer: caso faça perguntas que fujam do padrão apresentado nos exemplos, é possível que o *retriever* não consiga formar uma consulta coerente e, consequentemente, nenhum dado é recolhido nem retornado.

9.4 Testes e validação

Para a realização dos testes e validação, foi criada uma segunda instância do banco de dados, utilizada exclusivamente para experimentos relacionados ao comportamento do RAG. Nessa instância, foram inseridos apenas dados fictícios e situações simuladas, garantindo que nenhum dado real do banco principal fosse afetado durante o processo - especialmente em caso de falhas.

Os *scripts* ETL já haviam sido previamente testados para assegurar a correta alimentação do banco principal. Assim, nesta etapa, o foco dos testes foi sobre o funcionamento do RAG e foram conduzidos testes voltados para as respostas em formato JSON. Os testes seguiram as questões de competência anteriormente mencionadas da seguinte maneira: cada caso de teste avalia a resposta, em JSON, de um questão de competência sobre os dados fictícios. Cada teste avalia se os valores do JSON são correspondentes com o esperado. Por exemplo, caso o JSON represente um caminho entre duas entidades, é avaliado o(s) grau(s) de precisão de cada aresta retornada.

Por exemplo, para a pergunta "Onde o Aécio Neves da Cunha trabalha?"sobre o banco principal é gerado o seguinte texto JSON:

{ "pergunta_realizada": "ONDE O AÉCIO NEVES DA CUNHA TRABALHA?",

"cypher_gerado": "cypher
nMATCH (p:Pessoa)-[r:OCUPA]->(n:Organizacao) WHERE p.nome CONTAINS ’AÉCIO NEVES DA CUNHA’ RETURN

p, r, n ORDER BY r.grau_precisao
n",
"resultados": [
{
"registro_id": 1,
"colunas": {
"p": {
"tipo_retorno": "Node",
"labels": [
"Pessoa"
],
"propriedades": {
"grau_precisao": 5,
"cpf": "***.289.837-**",
"nome": "AÉCIO NEVES DA CUNHA"
},
"element_id": "4:fee7ee54-c710-4c93-b19e-3a57353eb2ac:1530"
},
"r": {
"tipo_retorno": "Relationship",
"tipo": "OCUPA",
"propriedades": {
"data_inicio": "2019-02-01T00:00:00",
"grau_precisao": 5,
"data_fim": "2023-01-31T00:00:00",

"cargo": "Deputado Federal"

},

"element_id": "5:fee7ee54-c710-4c93-b19e-3a57353eb2ac:11529215046068485 06",

"de": "AÉCIO NEVES DA CUNHA",

"para": "Câmara dos Deputados"

},

"n": {

"tipo_retorno": "Node",

"labels": [

"Organizacao"

],

"propriedades": {

"grau_precisao": 5,

"nome": "Câmara dos Deputados"

},

"element_id": "4:fee7ee54-c710-4c93-b19e-3a57353eb2ac:1379"

} } }] }

Neste formato normalizado, é possível ver que o JSON retorna três componentes principais, dentro da seção "resultados/colunas". Os componentes "p", "r"e "n"são nomenclaturas para definir entidades ou relacionamentos na *query*. Aqui, "p"representada a pessoa dada (Aécio Neves), "r"representa o relacionamento (do tipo "OCUPA") e "n"representa a entidade buscada (Organizacao onde Aécio Neves trabalha).

Com isso, os seguintes testes foram implementados:

1. test_onde_pessoa_trabalha:

- Se baseia na questão de competência "Onde a pessoa X trabalha"?;
- Analisa o nome do nó retornado, o qual seria a Organizacao onde uma pessoa dada trabalha.

2. test_pessoas_trabalham_em_x:

- Se baseia na questão de competência "Que pessoas trabalham em X"?;
- Analisa o nome do nó retornado, o qual seria a(s) Pessoa(s) que trabalha(m) em X.

3. test_irmaos_de_pessoa:

- Se baseia na questão de competência "Quem são os irmãos da pessoa X"?;
- Analisa o nome do nó retornado, o qual seria a(s) Pessoa(s) que é(são) irmão(s) de uma pessoa dada.

4. test_ligacao_entre_duas_pessoas:

- Se baseia na questão de competência "Qual é a ligação/relacionamento entre as pessoas X e Y?";
- Analisa o grau de precisão da relação entre duas pessoas dadas.

5. test_parentes_de_pessoa:

- Se baseia na questão de competência "Quais são os parentes da pessoa X?";
- Analisa o nome do nó retornado, o qual seria a(s) Pessoa(s) que é(são) parente(s) de uma pessoa dada.

6. test_irmaos_de_pessoa_em_empresa:

- Se baseia na questão de competência "Quais são os irmãos da pessoa X que trabalham na empresa Y?";

- Analisa o nome do nó retornado, o qual seria a(s) Pessoa(s) que trabalha(m) em uma dada organizacao e que é(são) irmão(s) de uma pessoa dada.

10
Considerações Finais

Para as considerações finais a serem consideradas, voltaremos a abordar sobre os objetivos deste trabalho e sobre possíveis trabalhos futuros.

10.1
Objetivos

Os objetivos principais deste trabalho foram definidos como a definição, construção e disponibilização de uma KB sobre PEPs e relacionamentos familiares e comerciais entre elas, além de possibilitar perguntas de linguagem natural para consultas. A alimentação da KB seria de forma automática e manual, com diferentes graus de precisão e as consultas ao banco seriam respondidas em JSON ou em LN.

A KB foi definida e modelada com inspiração em ontologias, remodelada para a implementação final e preenchida com dados familiares, laborais e comerciais de PEPs. Com isso, podemos dizer que o primeiro objetivo deste projeto foi concluído. Além disso, a dinâmica RAG e suas instâncias de LLM possibilitam perguntas de linguagem natural por parte do usuário, respostas do sistema em linguagem natural ou em JSON. Estes objetivos foram concluídos.

Porém, a alimentação da KB se deu apenas de forma automática, sem que usuários possam manipular o banco de dados. Mesmo que os relacionamentos entre entidades do banco tenham graus de precisão, eles não podem ser atribuídos por usuários. O objetivo de formas diferentes de alimentação do banco serão incluídas nos trabalhos futuros.

Por fim, por conta do grande escopo deste trabalho, mencionamos anteriormente sobre a substituição da API que retornaria a resposta em JSON por uma *feature* em *checkbox*. Assim, o JSON é retornado na própria interface. Por conta disto, mesmo que o sistema tenha provado que é capaz de retornar formatos JSONs, o objetivo de criação e implementação de uma API não foi

atingido.

10.2 Trabalhos Futuros

Com todas as limitações mencionadas nos diferentes componentes do sistema, o principal objetivo das futuras melhorias e incrementos é superar os atuais obstáculos. Uma implementação com um modelo de LLM mais capaz é um dos primeiros passos para que o sistema, no futuro, saia da definição de 'prova de conceito' e passe a ser um sistema totalmente funcional. Isso ajudará, como mencionado anteriormente, na grande parte das limitações hoje para a captura de dados.

Além disso, o uso de ferramentas para automatização e otimização do pipeline ETL de dados públicos sobre PEPs também seria um outro grande avanço para o sistema.

Um importante aspecto que ajudaria na investigação de fraudes seriam certas *features* em relação aos nomes de Pessoas e Organizações. Uma das *features* pensadas foi a de acompanhar - ou seja, realizar um *tracking* - sobre mudanças de nome dessas entidades que tentem encobrir certas atividades com nomes anteriores.

O registro de fontes de dados e datas de registro de nós e relacionamentos, tanto na modelagem quanto no banco de dados, também está considerado para passos futuros. Este incremento na estrutura do banco ajuda a ter maior transparência e proveniência dos dados inseridos.

A recuperação - ou disponibilização através de terceiros - de dados sobre sociedades, seus sócios e seus respectivos percentuais também se encaixa como incremento futuro a ser considerado para melhor aproveitamento das diferentes relações comerciais entre PEPs.

Por fim, novas questões de competência foram sugeridas pelo pesquisador Bruno Bondarowski e que estão consideradas para ajudar em incrementos deste sistema. Estas novas questões ajudariam a recolher informações interessantes

para potenciais investigações. São elas:

1. "Da lista de empresas contratadas pela cidade X, quais pessoas que tem relação com o prefeito Y foram beneficiadas?";
2. "A organização X, que recebeu Y milhões de emendas, tem quais relações com políticos/PEPs?"
3. "Liste os políticos que tem familiares contratados em governos citando qual a posição de cada um."

11
Referências bibliográficas

AL-RUITHE, M.; BENKHELIFA, E.; HAMEED, K. A systematic literature review of data governance and cloud data governance. **Personal and ubiquitous computing**, Springer, v. 23, n. 5, p. 839–859, 2019. Acessado em: 3 de Setembro de 2025.

ATLASSIAN. **What is a knowledge base?** 2025. Acessado em: 12 de Novembro de 2025. Disponível em: https://www.atlassian.com/itsm/knowledge-management/what-is-a-knowledge-base.

BARBOSA, G. de C. Transferências especiais: "pix" para quem? **CADERNOS DE FINANÇAS PÚBLICAS**, v. 25, n. 1, dez. 2024. Disponível em: https://publicacoes.tesouro.gov.br/index.php/cadernos/article/view/264.

BEZERRA, C.; FREITAS, F.; SANTANA, F. Evaluating ontologies with competency questions. In: **2013 IEEE/WIC/ACM International Joint Conferences on Web Intelligence (WI) and Intelligent Agent Technologies (IAT)**. [S.l.: s.n.], 2013. v. 3, p. 284–285. Acessado em: 27 de Agosto de 2025.

BRASIL, B. C. D. **CIRCULAR Nº 3.461**. 2009. Acessado em: 12 de Novembro de 2025. Disponível em: https://www.bcb.gov.br/pre/normativos/circ/2009/pdf/circ_3461_v4_p.pdf.

BROUS, P.; JANSSEN, M.; VILMINKO-HEIKKINEN, R. Coordinating decision-making in data management activities: a systematic review of data governance principles. In: SPRINGER. **International Conference on Electronic Government**. [S.l.], 2016. p. 115–125.

CARNUT, L. et al. Emendas parlamentares em saúde no contexto do orçamento federal: entre o 'é' e o 'dever ser' da alocação de recursos. **Saúde em Debate**, Centro Brasileiro de Estudos de Saúde, v. 45, n. 129, p. 467–480, Apr 2021. ISSN 0103-1104. Disponível em: https://doi.org/10.1590/0103-1104202112917.

DEPUTADOS, C. dos. **API Dados Abertos da Câmara dos Deputados**. 2025. Interface Swagger da API RESTful com acesso a dados legislativos como deputados, proposições, votações, etc. Acessado em 23 de Junho de 2025. Disponível em: https://dadosabertos.camara.leg.br/swagger/api.html.

DISNEY, A. **The ultimate guide to creating graph data models**. 2020. Acessado em: 12 de Novembro de 2025. Disponível em: https://cambridge-intelligence.com/graph-data-modeling-101/#:~:text=What%20is%20graph%20data%20modeling,and%20end-users%20much%20easier.

DOGAN, C. **Temporal Knowledge Graphs: Unlocking the Power of Time in Data**. 2024. Acessado em: 12 de Novembro de 2025. Disponível em: https://medium.com/self-study-notes/temporal-knowledge-graphs-unlocking-the-power-of-time-in-data-6c6c6bcde6a2.

ELEITORAL, T. S. **Portal de Dados Abertos do TSE**. 2025. Portal público com API para acesso a cerca de 164 conjuntos de dados eleitorais, substituto do Repositório de Dados Eleitorais (RDE) descontinuado em janeiro de 2022. Acessado em: 13 de Novembro de 2025. Disponível em: https://dadosabertos.tse.jus.br/.

GAUTAM, S. **Neo4j GraphRAG Workflow with LangChain and LangGraph**. 2024. Acessado em: 22 de Agosto de 2025. Disponível em: https://neo4j.com/blog/developer/neo4j-graphrag-workflow-langchain-langgraph/.

GOV.BR. **Nepotismo**. 2025. Disponível em: https://www.gov.br/cgu/pt-br/assuntos/prevencao-da-corrupcao/nepotismo.

GOV.BR. **O que são Pessoas Expostas Politicamente (PEPs)?** 2025. Disponível em: https://www.gov.br/coaf/pt-br/assuntos/informacoes-as-pessoas-obrigadas/o-que-sao-pessoas-expostas-politicamente-peps.

JØSANG, A.; ISMAIL, R.; BOYD, C. A survey of trust and reputation systems for online service provision. **Decision support systems**, Elsevier, v. 43, n. 2, p. 618–644, 2007.

JURíDICOS, P. da República Secretaria-Geral Subchefia para A. **LEI Nº 14.133, DE 1º DE ABRIL DE 2021**. 2021. Acessado em: 12 de Novembro de 2025. Disponível em: https://www.planalto.gov.br/ccivil_03/_ato2019-2022/2021/lei/l14133.htm.

KNEZ, T.; ŽITNIK, S. Event-centric temporal knowledge graph construction: A survey. **Mathematics**, v. 11, n. 23, 2023. ISSN 2227-7390. Disponível em: https://www.mdpi.com/2227-7390/11/23/4852.

LEITãO, S. P.; FORTUNATO, G.; FREITAS, A. S. d. Relacionamentos interpessoais e emoções nas organizações: uma visão biológica. **Revista de Administração Pública**, Fundação Getulio Vargas, v. 40, n. 5, p. 883–907, Sep 2006. ISSN 0034-7612. Disponível em: https://doi.org/10.1590/S0034-76122006000500007.

LINKURIOUS. **Graph data modeling: A quick guide**. 2025. Acessado em: 12 de Novembro de 2025. Disponível em: https://linkurious.com/graph-data-modeling/.

LIU, P. et al. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. **ACM Computing Surveys**, ACM, v. 55, n. 9, p. 1–35, 2023.

LIVRE, W. A. enciclopédia. **Base de conhecimento**. 2023. Acessado em: 12 de Novembro de 2025. Disponível em: https://pt.wikipedia.org/wiki/Base_de_conhecimento.

MORAIS, E. A. M.; AMBRóSIO, A. P. L. **Ontologias: conceitos, usos, tipos, metodologias, ferramentas e linguagens**. 2007. Acessado em 21 de Novembro de 2025. Disponível em: https://ww2.inf.ufg.br/sites/default/files/uploads/relatorios-tecnicos/RT-INF_001-07.pdf.

MOREIRA, V. R. Ontologias em sistemas multi-agentes. **Universidade Estadual de Campinas, UNICAMP**, 2010.

NEO4J, I. **Natural Language Queries — NeoDash User Guide**. 2025. Acessado em: 18 de Agosto de 2025. Disponível em: https://neo4j.com/labs/neodash/2.4/user-guide/extensions/natural-language-queries/.

NIELSEN, O. B. A comprehensive review of data governance literature. 2017.

NOY, N. F.; HAFNER, C. D. The state of the art in ontology design: A survey and comparative review. **AI Magazine**, v. 18, n. 3, p. 53, Sep. 1997. Acessado em: 27 de Agosto de 2025. Disponível em: https://ojs.aaai.org/aimagazine/index.php/aimagazine/article/view/1306.

ONTOLOGIES, S. **University Ontology (draft)**. 2000. Online; accessed 18-fevereiro-2025. Disponível em: https://www.cs.umd.edu/projects/plus/SHOE/onts/univ1.0.html.

OPENSANCTIONS. **OpenSanctions**. 2025. Disponível em: https://www.opensanctions.org/.

PICKLER, M. E. V. Web semântica: ontologias como ferramentas de representação do conhecimento. **Perspectivas em Ciência da Informação**, Escola de Ciência da Informação da UFMG, v. 12, n. 1, p. 65–83, Jan 2007. ISSN 1413-9936. Disponível em: https://doi.org/10.1590/S1413-99362007000100006.

POSTMAN, I. **Postman Flows**. 2025. Acessado em: 12 de Novembro de 2025. Disponível em: https://www.postman.com/product/flows/.

PúBLICO, C. N. do M. **Isonomia**. 2015. Acessado em: 12 de Novembro de 2025. Disponível em: https://www.cnmp.mp.br/portal/glossario/8017-isonomia.

RAJEBAH, N.; AL-KHALIFA, H. Semantic relationship extraction and ontology building using wikipedia: A comprehensive survey. **International Journal of Computer Applications**, v. 12, 12 2010.

REDESIM. **Consulta CNPJ Redesim**. 2018. Acessado em: 13 de Novembro de 2025. Disponível em: https://consultacnpj.redesim.gov.br/.

TEAM, N. D. **APOC Core Documentation (version 2025.05)**. 2025. Https://neo4j.com/docs/apoc/current/. Documentação oficial da biblioteca APOC Core para Neo4j, com introdução, instalação, procedimentos e funções em Cypher. Acessado em 23 de Junho de 2025. Disponível em: https://neo4j.com/docs/apoc/current/.

UNIãO, C.-G. da. **API de Dados – Portal da Transparência**. 2025. Https://portaldatransparencia.gov.br/api-de-dados. API pública com dados governamentais sobre despesas, receitas, convênios, obras, entre outros. Acessado em 23 de Junho de 2025. Disponível em: https://portaldatransparencia.gov.br/api-de-dados.

WATCHER, A. **AML Watcher**. 2025. Acessado em: 12 de Novembro de 2025. Disponível em: https://amlwatcher.com/.

Wikimedia Foundation. **API: Tutorial (pt-br)**. 2025. Https://www.mediawiki.org/wiki/API:Tutorial/pt-br. Tutorial oficial da API do MediaWiki em português para iniciantes, com exemplos de consulta em HTTP e formatos de resposta (JSON, XML, YAML). Acessado em: 23 de Junho de 2025. Disponível em: https://www.mediawiki.org/wiki/API:Tutorial/pt-br.

YANG, Y.; CHEN, J.; XIANG, Y. A review on the reliability of knowledge graph: from a knowledge representation learning perspective. **World Wide Web**, v. 28, n. 4, 2025. Acessado em: 31 de Agosto de 2025. Disponível em: https: //doi.org/10.1007/s11280-024-01316-w.

ZAFAR, F. et al. Trustworthy data: A survey, taxonomy and future trends of secure provenance schemes. **Journal of Network and Computer Applications**, v. 94, p. 50–68, 2017. ISSN 1084-8045. Disponível em: https://www.sciencedirect.com/ science/article/pii/S1084804517302229.