Universidade Federal de Juiz de Fora
Instituto de Ciências Exatas
Bacharelado em Ciência da Computação

Integração e avaliação de modelos de atividade docente no magistério superior

Daniel Machado Barbosa Delgado

JUIZ DE FORA
SETEMBRO, 2024

Integração e avaliação de modelos de atividade docente no magistério superior

Daniel Machado Barbosa Delgado

Universidade Federal de Juiz de Fora

Instituto de Ciências Exatas

Departamento de Ciência da Computação

Bacharelado em Ciência da Computação

Orientador: Igor de Oliveira Knop

JUIZ DE FORA

SETEMBRO, 2024

Integração e avaliação de modelos de atividade docente no magistério superior

Daniel Machado Barbosa Delgado

MONOGRAFIA SUBMETIDA AO CORPO DOCENTE DO INSTITUTO DE CIÊNCIAS EXATAS DA UNIVERSIDADE FEDERAL DE JUIZ DE FORA, COMO PARTE INTEGRANTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE BACHAREL EM CIÊNCIA DA COMPUTAÇÃO.

Aprovada por:

Igor de Oliveira Knop
D.Sc. Modelagem Computacional

Edmar Welington Oliveira
D.Sc. Informática

André Luiz de Oliveira
D.Sc. Ciência da Computação

JUIZ DE FORA
27 DE SETEMBRO, 2024

Resumo

Durante o ano letivo, os professores do magistério superior desenvolvem atividades em diferentes eixos de educação. Em conformidade com o artigo 3º da Lei nº 14.129/2021, que estabelece diretrizes para garantir a transparência na execução dos serviços públicos e o monitoramento da qualidade desses serviços, os docentes devem preencher o Relatório Individual de Trabalho (RIT) com o andamento das atividades. Existem outras fontes de dados de atividade que podem ser levadas em consideração para que a Comissão Permanente de Avaliação Docente (CPAD) consiga avaliar o desempenho docente e averiguar se suas atividades estão de acordo com seu plano individual de trabalho. Este trabalho de conclusão de curso consiste em explorar o uso de ferramentas de bases de conhecimento e inteligência artificial generativa aplicadas na avaliação docente em relação às suas atividades. A importação das atividades realizadas populam um banco de dados em grafos que serve para melhorar a geração na recuperação na forma de um serviço de perguntas, baseado em um grande modelo de linguagem. Dessa forma, espera-se permitir uma melhor gestão acadêmica, explorando situações de preenchimento incorreto e explorar a criação de novas métricas de avaliação, garantindo uma análise mais assertiva e com dados mais precisos.

Palavras-chave: desenvolvimento de software; gestão acadêmica; inteligência artificial; bases de conhecimento.

Abstract

During the academic year, higher education faculty members engage in various educational activities. In accordance with Article 3 of Law No. 14,129/2021, which establishes guidelines to ensure transparency in the execution of public services and the monitoring of their quality, faculty members are required to complete the *Individual Work Report* (IWR) to document the progress of their activities. However, there are additional sources of data that can be used by *Permanent Committee of Teaching Activity* (PCOTA) to assess teaching performance and ensure the quality of education. This project explores the use of knowledge base tools and generative artificial intelligence applied to faculty performance evaluation concerning their activities. The importation of performed activities populates a graph database, which enhances question-based retrieval through a large language model service. Thus, this approach aims to improve academic management by addressing incorrect entries and introducing new evaluation metrics, ensuring a more accurate analysis with precise data.

Keywords: software development; academic management; artificial intelligence; knowledge base.

Agradecimentos

A Deus, família, amigos e professores que me guiaram até aqui.

Conteúdo

Lista de Figuras

Lista de Abreviações e Siglas

API *Application Programming Interface*. 18, 42

APOC *Awesome Procedures On Cypher*. 39

BDOG Banco de Dados Orientado a Grafos. 19, 20, 39

CEPE Conselho de Ensino, Pesquisa e Extensão. 11

CNPq Conselho Nacional de Desenvolvimento Científico e Tecnológico. 18

CPAD Comissão Permanente de Avaliação Docente. 1, 17

CSV *Comma-separated Values*. 6, 37

DCC Departamento de Ciência da Computação. 18, 32, 37

GDS *Graph Data Science*. 6, 37, 47

IA Inteligência Artificial. 20, 22, 23, 34

ICE Instituto de Ciências Exatas. 17

IFET Instituto Federal de Educação, Ciência e Tecnologia. 29, 30

IWR *Individual Work Report*. 2

JSON *JavaScript Object Notation*. 6, 35, 37, 38, 41

LLM *Large Language Model*. 6, 12, 13, 16, 23, 24, 28, 29, 34, 36, 42, 43, 53

MEC Ministério da Educação. 17, 30

NoSQL *Not Only SQL*. 19

PCOTA *Permanent Committee of Teaching Activity*. 2

PIT Plano Individual de Trabalho. 11, 17

RAG *Retrieval-augmented generation*. 6, 13, 16, 23, 34, 36, 41, 53, 54

RDF *Resource Description Framework*. 28

RIT Relatório Individual de Trabalho. 1, 6, 11, 12, 17, 18, 32, 33, 35, 37–39, 41, 45

SGBD Sistema de Gerenciamento de Banco de Dados. 32

SIAPE Sistema Integrado de Administração de Recursos Humanos. 37

SQL *Structed Query Language*. 24, 28, 29

SUS *System Usability Scale*. 55

TOPSIS *Technique for Order Preference by Similarity to Ideal Solution*. 30

UFJF Universidade Federal de Juiz de Fora. 11, 12, 14, 17, 18, 35, 37, 45

XML *Extensible Markup Language*. 18

1 Introdução

1.1 Apresentação do tema

A educação é um pilar para o desenvolvimento de uma sociedade, especialmente no que se refere ao ensino superior. É através das universidades que são formados profissionais capacitados e aptos a enfrentar os desafios do mercado de trabalho, além de ser um espaço para o desenvolvimento de pesquisas e produção científica que impulsionam a inovação e o progresso.

Além disso, a educação superior também tem um papel crucial na formação de cidadãos críticos e conscientes de seu papel na sociedade. Por meio de uma educação de qualidade, os indivíduos são capacitados a compreender o mundo ao seu redor, refletir sobre questões sociais, políticas e econômicas, e atuar de forma proativa e responsável na transformação da realidade (CAPLAN, 1974).

A qualidade do ensino está diretamente relacionada ao desempenho dos professores (WHITEBOOK, 2003), que devem possuir conhecimentos técnicos, teóricos e práticos em sua área de atuação, além de habilidades pedagógicas e didáticas que permitam a transmissão desses conhecimentos de forma clara e objetiva aos estudantes.

Contudo, avaliar a atividade docente no ensino superior ainda é um grande desafio para as instituições de ensino (WANKE et al., 2021). É necessário coletar e analisar uma série de dados, como a produção científica e acadêmica dos professores, a participação em atividades de extensão e pesquisa, entre outros indicadores relevantes. Coletar todos os dados necessários, de forma precisa e organizada e fazer um acompanhamento de seus índices para melhor atender as finalidades da instituição, não é uma tarefa fácil. E mais difícil ainda, é disponibilizá-los de forma estruturada e coesa para os gestores tomarem decisões em suas instituições.

1.2 Contextualização

Um modelo de avaliação de atividade docente tem o objetivo de recolher e analisar informações relativas à competência, ao desempenho e à eficácia dos professores. É considerada essencial para que estes, através de processos adequados de formação e de desenvolvimento profissional, possam aprimorar suas práticas pedagógicas e consequentemente a produção acadêmica nas instituições.

Para realizar uma avaliação completa e precisa, é necessário que as instituições de ensino possuam uma estrutura adequada, com sistemas informatizados e ferramentas de análise de dados capazes de fazer um acompanhamento constante dos índices de desempenho dos professores e da qualidade do ensino oferecido.

Na Universidade Federal de Juiz de Fora (UFJF), é obrigatório que todos os professores apresentem um Plano Individual de Trabalho (PIT) em conformidade com a Resolução 46/95 do Conselho de Ensino, Pesquisa e Extensão (CEPE) da universidade. O PIT é um documento elaborado por cada docente e deve conter um detalhamento das atividades planejadas para o próximo ano. Ao final do período, um RIT é produzido para detalhar as atividades realizadas e os resultados alcançados. A chefia imediata do professor, assim como a assembleia docente avaliam se as atividades estão de acordo com o PIT e se sua produção está alinhada com o plano de carreira, com base nesses relatórios.

1.3 Descrição do problema

O método de avaliação do RIT através do modelo atual, baseado em pesos fixos, pode não corresponder à realidade dos departamentos de forma precisa, o que pode comprometer a exatidão da avaliação do desempenho dos professores. Tratando-se de uma instituição de ensino público, é notório que, com o passar dos anos, as prioridades institucionais mudam, os recursos para pesquisa e extensão variam e, portanto, trabalhar com modelos de avaliação docente rígidos não é um boa escolha. Essa situação se distancia do ideal definido por (FERNANDES, 2008), que reitera a principal característica desse tipo de modelo, que é a utilização de procedimentos para medir o desempenho docente da maneira mais precisa possível.

Além do RIT, existem plataformas que contém dados importantes para a avaliação docente como o *Lattes* e o *Siga* além de publicações online. Para fazer a utilização desses dados, é necessário passar por diversos desafios. O primeiro é o de coletar os dados, pois a UFJF não possui uma ferramenta automatizada que seja capaz de trazer esses dados desejados. Outra barreira é a integração dos dados, pois cada plataforma possui um modelo diferente para exibir as informações, e utilizar mais de uma fonte de dados não é simples.

Surge então o problema da integração dos dados, pois há a necessidade de combinar e integrar dados de diferentes fontes para criar um modelo unificado e consistente. As fontes de dados podem variar em termos de formato, estrutura e conteúdo, o que pode tornar difícil a tarefa de unificar esses dados em um modelo coeso e padronizado. Somado a isso, temos o problema da qualidade dos dados que constam no RIT. Dado que o preenchimento é feito manualmente, esse processo está sujeito a erros, logo, ao fazer análises em um conjunto possivelmente impreciso, podem ser geradas falsas conclusões.

1.4 Justificativa e Motivação

Uma instituição de ensino que possui bons métodos de avaliação dos docentes, é capaz de fazer ajustes para criar um ambiente que favoreça a melhora do desempenho dos professores, e por consequência, a qualidade do ensino também é favorecida (WHITEBOOK, 2003).

Para prover uma melhor forma de avaliar as atividades docentes, é necessário superar alguns desafios técnicos que ocorrem durante a aplicação de um método informatizado, como a duplicação de dados, conflitos de nomenclatura, inconsistência de dados, variações nos formatos de dados e a falta de correspondência entre diferentes fontes. Essas situações são corriqueiras nesse tipo de problema, e podem aumentar a complexidade da tarefa. Em cenários como esse, organizações implementam soluções com grafos de conhecimento (ALIYU; KANA; ALIYU, 2020) , técnicas de inteligência artificial envolvendo LLMs, *fine-tuning* e *prompt engineering* (ZHOU; ZHAO; LI, 2024).

Com um grafo de conhecimento, é possível criar uma ontologia que define as relações entre as diferentes entidades e como elas se relacionam (WANG et al., 2024).

Podendo identificar entidades duplicadas e relacionar entidades semelhantes de diferentes fontes de dados.

Logo, com o tratamento desses dados e a criação de um modelo único que seja flexível o suficiente para compreender essas diferenças, considerando a mutabilidade dos processos e os critérios dos departamentos, tornaria possível retirar várias métricas para contribuir com o planejamento estratégico da instituição.

A forma com que se interage com o sistema informatizado também é muito importante. Uma aplicação que consegue receber entradas de texto em linguagem natural e interpretá-las é boa para o usuário. LLMs têm se mostrado uma boa aliada para esse tipo de problema, pois são capazes de interpretar consultas e fornecer respostas com base nas informações fornecidas. Esses modelos conseguem processar grandes quantidades de dados e aprender padrões linguísticos, facilitando a interação entre humanos e sistemas computacionais, o que melhora a experiência do usuário ao permitir uma comunicação mais intuitiva e eficiente (KUM; KIM; LEE, 2023).

No entanto, os LLMs apresentam limitações ao lidar com contextos de dados específicos para os quais não foram previamente treinados. Diante dessas restrições de domínio, os grafos de conhecimento podem ser integrados com os LLMs. Utilizando uma abordagem de RAG é possível recuperar informações relevantes e gerar respostas baseadas em um contexto específico, como, por exemplo, fornecendo o esquema de um banco de dados ao LLM para melhorar a precisão e relevância das respostas geradas (PI, 2024).

1.5 Objetivos

Neste contexto, o objetivo geral deste trabalho é realizar uma gestão mais eficiente dos dados acadêmicos, visando melhorar a gestão acadêmica nos departamentos do ensino superior. Dessa forma, podem ser elencados os seguintes objetivos específicos:

1. Realizar um mapeamento das principais métricas colhidas em um departamento real, como estudo de caso;

2. Propor um sistema de raspagem de dados (*crawling*) no meio atual de coleta de dados para recuperação dessas informações;

3. Definir um modelo em grafo de conhecimento, que permita modelar os dados das fontes identificadas;

4. Construir um sistema de geração aumentado de recuperação de dados por grafos.

5. Definir um conjunto de métricas que permita quantificar produção, esforço despendido e contribuição para o planejamento estratégico institucional;

6. Propor ferramentas e políticas de identificação de erros de preenchimento e inconsistências.

A realização desses objetivos específicos resultou em novas ferramentas que se mostraram benéficas para melhorar a precisão da medição do desempenho docente no magistério superior da UFJF. O uso dessas ferramentas permitirá que a instituição avalie de forma mais eficaz a qualidade do ensino oferecido aos alunos, garantindo assim um ambiente acadêmico de excelência.

1.6 Metodologia

Neste trabalho é apresentada uma pesquisa exploratória que busca permitir uma melhor gestão acadêmica através de uma avaliação mais precisa das atividades docentes no magistério superior com o usado de geração aumentada de recuperação de dados baseada em grafos. Primeiramente, foi criado um modelo de dados flexível capaz de integrar todas as informações relevantes para a avaliação docente. Em seguida, empregaremos técnicas sistemáticas e automatizadas para coletar dados reais e preencher o banco de dados. Posteriormente, definiremos métricas para avaliar o desempenho dos docentes e conduziremos uma análise minuciosa desses resultados, comparando-os com o estado atual. Como passos metodológicos estão previstos os seguintes:

1. Mapeamento das ferramentas, leis e normas que regem a avaliação docente;

2. Montagem técnica da integração dos modelos;

3. Criação de modelos dos dados de integração;

4. Obtenção de dados reais por raspagem de dados;

5. Alimentação da base de dados com dados reais;

6. Implementação do sistema de geração aumentada de recuperação de dados baseada em grafos.

7. Definição de métricas e inconsistência nos dados;

8. Avaliação dos resultados e anonimamento da base de dados para publicação.

1.7 Organização deste trabalho

Além desta Introdução, este trabalho está organizado em cinco capítulos. O Capítulo 2 apresenta os principais conceitos necessários para o entendimento da solução proposta. O Capítulo 3 apresenta os trabalhos relacionados que foram utilizados para o desenvolvimento desse projeto. O Capítulo 4 apresenta todo o desenvolvimento do projeto. Por fim, o Capítulo 5 compila os resultados e apresenta as limitações e trabalhos futuros.

2 Fundamentação Teórica

A avaliação das atividades docentes no magistério superior é um processo complexo e importante que envolve diferentes aspectos do trabalho do professor. Nesse contexto, é essencial compreender os conceitos e as modalidades de avaliação disponíveis, bem como as estratégias técnicas utilizadas no processo e os indicadores de análise que podem ser utilizados para tomar decisões mais informadas. O presente capítulo é destinado a apresentar os conceitos que são fundamentais e fornecer o conhecimento teórico necessário para o entendimento dos temas abordados nesse trabalho.

Na Seção 2.1, são apresentados os conceitos relacionados ao magistério no ensino superior, abordando suas características e particularidades. Em seguida, a Seção 2.2 explora as atividades desenvolvidas pelos docentes. Na Seção 2.3, são apresentadas as principais fontes de dados das atividades docentes. A Seção 2.4 introduz os conceitos de bancos de dados orientados a grafos, explicando sua estrutura e funcionamento, enquanto a Seção 2.5 trata dos grafos de conhecimento, destacando sua importância para modelagem de dados complexos. Na Seção 2.6, são discutidas as possibilidades de análises em redes complexas buscando interações entre os docentes. A Seção 2.7 traz uma visão sobre a inteligência artificial, com destaque para as utilizações de ia generativa, LLM, RAG e *engenharia de prompt*. Por fim, a Seção 2.8 apresenta a linguagem de consulta *Cypher*, essencial para a manipulação de dados no banco de grafos utilizado neste projeto.

2.1 Magistério no Ensino Superior

A educação superior tem, entre outras, a finalidade suscitar o desejo permanente de aperfeiçoamento cultural e profissional e possibilitar a correspondente concretização, integrando os conhecimentos que vão sendo adquiridos numa estrutura intelectual sistematizadora do conhecimento de cada geração (BRASIL, 1996, art.43). É condição para habilitação no Magistério Superior ao docente de nível superior ter concluído a formação em programas de pós-graduação, na modalidade de mestrado ou doutorado. Ao corpo

docente é atribuído atividades de extensão, planos de trabalho, programas e projetos de pesquisa científica. São atividades dos docentes do Magistério Superior a aplicação de ensino e pesquisa, exercendo funções de monitoria de acordo com rendimento e plano de estudos (BRASIL, 1996, art.84).

A qualidade dos serviços é essencial para o desenvolvimento tanto individual quanto institucional (REIFSCHNEIDER, 2008). Na busca da avaliação de desempenho, espera-se que o processo sistemático de coleta de dados se baseie em critérios pré-estabelecidos e conhecidos por aqueles que são avaliados, permita a formação de valores baseados em evidências. Seguindo as determinações do Ministério da Educação (MEC), a resolução (EXATAS, 2016) normatiza a elaboração, acompanhamento e avaliação do PIT e do RIT dos docentes, sendo a principal forma de planejamento e acompanhamento das atividades dos professores (EXATAS, 2016).

2.2 Atividades desenvolvidas pelos docentes

O processo de avaliação das atividades dos professores é crucial para uma boa gestão. De acordo com as normas estabelecidas pelo Instituto de Ciências Exatas (ICE) da UFJF em (EXATAS, 2016), a CPAD é responsável por avaliar o PIT e o RIT dos professores. O PIT é uma proposta que distribui o esforço do professor em cinco eixos: ensino, pesquisa, extensão, gestão e afastamento-capacitação. Já o RIT é um relato detalhado de cada atividade, acompanhado de documentos comprobatórios quando necessário.

Após a avaliação da CPAD, é emitido um parecer que verifica se o PIT corresponde ao plano de metas estabelecido pelo departamento. Além disso, o RIT é avaliado para verificar se está em consonância com o PIT apresentado pelo professor no início do ano. Todos os pareceres são submetidos à aprovação pela assembleia departamental. É importante destacar que esse processo é essencial para garantir a qualidade do ensino, pesquisa, extensão, gestão, afastamento e capacitação dos professores do Magistério Superior (EXATAS, 2016).

2.3 Fontes de dados de atividades desenvolvidas pelos docentes

Os dados e a análise dos mesmos é importante para toda e qualquer organização, uma vez que a tentativa de identificar especificidades pode trazer melhores condições para o desenvolvimento de novos estudos, com base num melhor entendimento conceitual do processo, alinhado aos paradigmas existentes (TEIXEIRA, 2003).

O RIT, no âmbito do Departamento de Ciência da Computação (DCC) é preenchido em planilhas *Google Sheets* que ficam compartilhadas. Os dados são coletados internamente de modo que cada professor preenche manualmente o seu respectivo RIT. O *Google Drive*, onde ficam armazenados, provê uma *Application Programming Interface* (API) para extrair as planilhas.

Outra fonte de dados é a plataforma *Lattes* , que representa a experiência do Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) na integração de bases de dados de Currículos, de Grupos de pesquisa e de Instituições em um único Sistema de Informações (LATTES, 2023). Pela quantidade diversa de informações, é uma fonte importante para extrair dados, que no caso pode ser feita via *web scrapping*, com um script escrito em *python* ou via arquivo *Extensible Markup Language* (XML).

A UFJF possui o SIGA, que é um sistema online que gerencia as informações acadêmicas e administrativas da universidade (UFJF, 2023). Na plataforma existem relatórios elaborados com informações importantes para o corpo docente, porém não possui uma API para fornecer esses documentos, e portanto, a forma de obtenção dos mesmos é via solicitação na central de atendimento.

Com isso, surge o problema da integração dos dados, e como mencionado por (RAHM; DO et al., 2000), quando múltiplas fontes de dados precisam ser integradas, a necessidade de utilizar estratégias de *data cleaning* para lapidar esses dados aumenta significativamente.

Para alcançar uma análise dos dados eficiente e precisa é necessário trabalhar com dados de qualidade e sem inconsistências. Como neste trabalho é necessário corrigir erros de digitação, o algoritmo de *Jaro-Winkler* pode ser uma ferramenta útil. Ele traz

como resultado uma medida de similaridade entre duas strings, que é utilizada para identificar a correspondência entre elas. Esse algoritmo também é utilizado em aplicações de processamento de linguagem natural para a correção de ortografia ou para agrupamento de strings. (WANG; QIN; WANG, 2017).

2.4 Banco de Dados Orientado a Grafos

Os bancos de dados orientados a grafos permitiram ultrapassar barreiras que existem no campo do modelo relacional. Essa alternativa surge para atender a necessidade de representar estruturas hierárquicas (TUIJN; GYSSENS, 1996).

Os bancos de dados orientados a grafos, como o Neo4j, permitiram ultrapassar barreiras que existem no campo do modelo relacional, oferecendo uma estrutura mais natural para capturar e analisar as relações entre dados (TUIJN; GYSSENS, 1996). Essa abordagem é útil para modelar estruturas hierárquicas e altamente conectadas, como redes sociais, sistemas de recomendação e análise de fraudes.

Essas aplicações se beneficiam da capacidade dos grafos de modelar conexões complexas entre entidades, algo que seria difícil e possivelmente menos eficiente em um banco de dados relacional. A flexibilidade no modelo de dados de um grafo permite que novas relações e propriedades sejam adicionadas sem grandes interrupções, tornando-os ideais para ambientes dinâmicos e iterativos.

Com a popularização das redes sociais e serviços de nuvem, novas propostas relacionados a bancos de dados têm ganho popularidade. Os bancos *Not Only SQL* (NoSQL), podem oferecer melhores desempenhos e escalabilidade em determinados cenários, devido à sua estrutura não relacional. No entanto, para obter essas melhorias, muitas vezes é necessário sacrificar a disponibilidade, a tolerância à partição de dados ou a consistência. Entre os bancos NoSQL, bancos orientados a grafos como o Neo4J ficam em evidência por ofertar consultas de maneira mais amigável para o desenvolvedor (ANGLES; GUTIERREZ, 2008).

Uma das características mais relevantes de um Banco de Dados Orientado a Grafos (BDOG) é que a topologia dos dados também carregam informações. Dessa maneira, as consultas podem ser feitas por navegação entre os nós do grafo de forma a avaliar

se o caminho percorrido satisfaz determinadas propriedades específicas (BONIFATI; CIUCANU; LEMAY, 2015).

Dentro das redes sociais existe uma interconexão massiva dos dados, como rede de pessoas, etiquetas (*tags*), atividades e comentários. Cenários como esse geram inúmeras relações *many to many*, relações essas que em bancos relacionais são extremamente custosas por conta das junções necessárias para manipular esses dados, trazendo então um desempenho insatisfatório. Entretanto, em um BDOG como essas relações são simples e de pouco custo, é possível entregar manipulações desse gênero com alto desempenho(AMMAR, 2016).

2.5 Grafos de Conhecimento

Os grafos de conhecimento estão presentes em sistemas de informação que requerem acesso a conhecimento estruturado. Eles podem ser dependentes ou independentes de domínio (PAULHEIM, 2017). Eles constituem um paradigma flexível de representação do conhecimento com a finalidade de descomplicar o processamento do conhecimento tanto para humanos quanto para máquinas. São vistos como um grande facilitador para vários casos de uso cada vez mais populares, incluindo pesquisa na web, resposta a perguntas e assistentes pessoais, além de permitir que outros aplicativos baseados em Inteligência Artificial (IA) sejam usados na maioria dos setores (DIRSCHL et al., 2020).

Devido à sua capacidade de representação versátil, os grafos de conhecimento podem ser usados para integrar diferentes fontes de dados heterogêneas, dentro de organizações (HEIST et al., 2020). Graças a esta função, eles estão se tornando uma poderosa ferramenta de gerenciamento de dados institucionais. Dentre as diversas formas de modelar um grafo de conhecimento, o grafo de propriedade é um dos modelos mais utilizados.

Um grafo de propriedade é um multigrafo rotulado direcionado com a especial característica que cada nó ou aresta pode manter um conjunto de pares propriedade-valor. Esse fato oferece flexibilidade adicional ao modelar dados (HOGAN et al., 2021). Do ponto de vista da modelagem de dados, um nó representa uma entidade, uma aresta representa um relacionamento entre entidades e uma propriedade representa uma característica específica de uma entidade ou relacionamento (ANGLES, 2018). Pode ser visto

na Figura 2.1 uma definição formal da noção abstrata descrita acima.

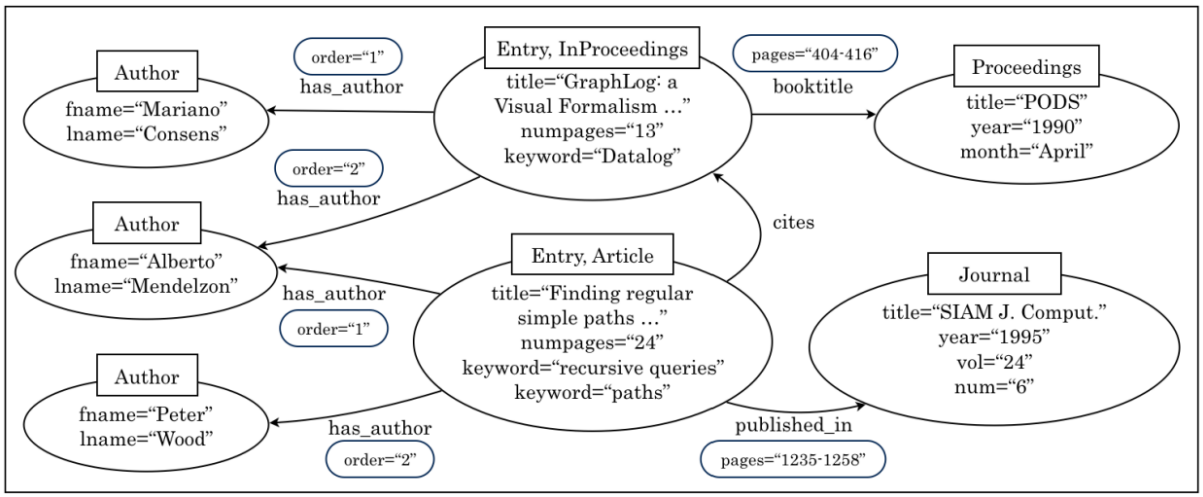

Figura 2.1: Exemplo de um grafo de propriedades representando informações bibliográficas. Parte do grafo ilustra o artigo de título "*Finding regular simple paths...*" com o relacionamento "*published_in*", mostrando que foi publicado no veículo de publicação de título "*SIAM J. Comput.*", e também mostra a ligação do artigo com o nó de autor, que possui as propriedades de primeiro nome e último nome, sendo "*Peter*" e "*Wood*", respectivamente. Fonte: Angles (2018)

.

2.6 Redes complexas

Redes complexas são estruturas que modelam sistemas compostos por muitos elementos interconectados de forma não trivial e são amplamente utilizadas para representar uma variedade de sistemas reais. Diferente das redes simples, onde as conexões são regulares ou aleatórias, as redes complexas exibem propriedades emergentes como distribuição de grau heterogênea, presença de comunidades, e robustez a falhas (NEWMAN, 2003).

Um aspecto relevante no estudo de redes complexas é a detecção de comunidades, que são grupos de nós que estão mais densamente conectados entre si do que com o restante da rede. A identificação dessas comunidades pode revelar importantes informações sobre a estrutura interna da rede, como grupos sociais em redes humanas ou módulos funcionais em redes biológicas. A detecção de comunidades permite entender melhor a organização da rede e pode ser utilizada para otimizar processos, como a difusão de informações ou a identificação de influenciadores (FORTUNATO, 2010).

Modularidade é uma métrica amplamente utilizada para avaliar a qualidade

da divisão de uma rede em comunidades. Essa medida quantifica o quanto uma dada divisão em comunidades é melhor do que uma divisão aleatória, levando em consideração o número de arestas dentro das comunidades em comparação ao número esperado em uma rede aleatória com a mesma distribuição de grau. Dessa forma, a modularidade é maior quando existem mais conexões dentro das comunidades do que seria esperado dentro de uma rede que foi randomizada, indicando então uma estrutura de comunidade bem definida (NEWMAN, 2006).

Existem diversos métodos que são utilizados para detectar comunidades, dentre eles, o algoritmo de Louvain se destaca pela sua eficiência na otimização da modularidade, onde a rede é particionada de forma a maximizar o ganho de modularidade em cada etapa. Operando em duas fases, sendo na primeira, realizada a atribuição para cada nó sua própria comunidade e depois os nós são movidos de uma comunidade para outra de forma iterativa, para maximizar o ganho de modularidade. Na segunda fase, uma nova rede é construída, onde as comunidades identificadas são agregadas em supernós, e o processo é repetido até que não haja mais ganhos significativos de modularidade. Esse processo hierárquico permite lidar com redes de grande escala de maneira eficiente (BLONDEL et al., 2008).

A aplicação do algoritmo de Louvain em redes complexas pode revelar hierarquias naturais de comunidades, permitindo uma compreensão mais profunda da estrutura interna dessas redes. Essa capacidade de revelar a organização modular das redes complexas torna o algoritmo uma ferramenta poderosa para análise em diversos domínios (FORTUNATO, 2010).

2.7 Inteligência Artificial

A IA é um ramo da computação que é dedicada à criação de máquinas e programas inteligentes. Inteligência, nesse contexto, é a capacidade de atingir objetivos que pode ser manifestada de diferentes formas tanto em pessoas, animais ou máquinas. No entanto, ainda não existe uma definição universal para inteligência que não dependa da comparação com a humana. Portanto, o campo da IA, foca em entender e replicar certos mecanismos de inteligência, permitindo que computadores realizem tarefas específicas de maneira

eficiente (MCCARTHY et al., 2007).

Dentro da área de IA, existe um campo específico chamado inteligência artificial generativa, que por sua vez, se refere a técnicas computacionais capazes de gerar novos conteúdos significativos, como texto, imagens ou áudio, a partir de dados de treinamento (FEUERRIEGEL et al., 2024). Exemplos amplamente difundidos de aplicações dessa tecnologia, são DALL-E 2, que é um sistema que tem a capacidade de criar imagens realistas a partir de uma descrição com linguagem natural, e o ChatGPT, que utiliza modelos de linguagem avançados para gerar respostas em texto que imitam conversas humanas de forma coerente e fluida.

Os LLM são modelos de IA projetados para entender, gerar e manipular texto em linguagem natural. Eles são treinados em enormes quantidades de dados textuais, o que permite capturar padrões, estruturas e contextos da linguagem humana. Utilizando técnicas avançadas de aprendizagem profunda, os LLMs conseguem realizar tarefas como tradução automática, sumarização de textos, respostas a perguntas e até mesmo a geração de textos criativos. LLMs são amplamente utilizados em assistentes virtuais, *chatbots* e aplicações que demandam interações baseadas em linguagem (CHANG et al., 2024).

Um conceito que é utilizado para aumentar o aproveitamento dos LLMs é a "engenharia de prompt": como os modelos de linguagem recebem uma entrada textual, esse conceito se trata da técnica estrutural elaborada que irá guiar o modelo de linguagem para produzir respostas de forma mais direcionada. Essa técnica faz ajustes no formato, no contexto e os detalhes da entrada para melhorar os resultados (REYNOLDS; MCDONELL, 2021).

Uma abordagem promissora que combina LLM com a recuperação de informações é o método RAG. Esse método aprimora a capacidade dos LLMs de gerar respostas precisas e contextualizadas ao combinar geração de texto com recuperação de dados relevantes de um conjunto externo. O sistema de RAG funciona em duas etapas: primeiro, ele recupera informações relevantes de uma base de dados ou documentos. Em seguida, essas informações são utilizadas para guiar o modelo de geração na criação de uma resposta mais informada e adequada à pergunta ou contexto fornecido. A combinação de geração e recuperação permite que sistemas de RAG superem limitações comuns dos

LLMs, como a obsolescência de dados e a falta de precisão em respostas que exigem um contexto específico (LEWIS et al., 2020).

2.8 Cypher

Cypher é uma linguagem de consulta declarativa desenvolvida para o banco de dados orientado a grafos Neo4j. Semelhante ao *Structed Query Language* (SQL), a linguagem *Cypher* permite que os usuários expressem consultas de maneira simples e intuitiva, lidando diretamente com nós, arestas e propriedades do grafo (NEO4J, 2024). Essa linguagem é utilizada no presente trabalho para a modelagem e consulta de dados relacionados às atividades acadêmicas.

Uma consulta básica em *Cypher* consiste em expressar padrões de nós e relacionamentos que o banco deve buscar, seguidos pela especificação dos dados a serem retornados. Um exemplo simples que busca todos os nós rotulados como "Docente"no grafo pode ser expresso como:

```
1 MATCH (d:Docente)
2 RETURN d
```

A cláusula *MATCH* é usada para localizar um padrão no grafo, e *RETURN* é usada para exibir os nós que correspondem à consulta. *Cypher* facilita a leitura e o entendimento das consultas, fazendo uso de uma sintaxe declarativa, o que contribui para sua expressividade e facilidade de uso (ROBINSON; WEBBER; EIFREM, 2013). Um exemplo mais claro da expressividade da linguagem pode ser visto a seguir :

```
1 MATCH (d:Docente)-[:REALIZA_ATIVIDADE]->(a:Atividade)
2 RETURN d.nome, a.descricao
```

Neste caso, o padrão de correspondência *-[:REALIZA_ATIVIDADE]-* ¿ descreve a relação entre o nó "Docente"e o nó "Atividade", retornando o nome do docente e a descrição da atividade realizada.

Há também a possibilidade de aplicar filtros e realizar operações de agregação, para refinar os resultados e gerar consultas mais complexas, como contar quantas atividades um determinado docente realizou:

```
1 MATCH (d:Docente)-[:REALIZA_ATIVIDADE]->(a:Atividade)
2 WHERE d.nome = 'Professor 1'
3 RETURN d.nome, count(a) AS total_atividades
```

Além dessas consultas, é possível a manipulação dos dados, como a inserção de um novo relacionamento entre dois docentes que coorientam um aluno, que seria :

```
1 MATCH (d1:Docente {nome: 'Prof. A'}),
2 (d2:Docente {nome: 'Prof. B'})
3 MERGE (d1)-[:CO_ORIENTA]->(d2)
```

No exemplo acima, a cláusula *MERGE* garante que o relacionamento "CO_ORIENTA"entre os dois docentes seja criado, caso ainda não exista. Essa funcionalidade é útil para evitar duplicidade de relacionamentos no grafo.

Com a simplicidade e expressividade da linguagem *Cypher*, é possível a realização da modelagem de relações complexas, como as que encontramos no cenário acadêmico. A flexibilidade oferecida para as consultas permite que as interações entre as entidades do projeto sejam exploradas de maneira eficiente, possibilitando uma análise profunda e significativa dos dados (ANGLES; GUTIERREZ, 2008).

2.9 Considerações Parciais

Neste capítulo, foram introduzidos os principais conceitos para o desenvolvimento deste trabalho, abordando desde o papel do magistério no ensino superior até as atividades desempenhadas pelos docentes e suas fontes de dados. Conceitos teóricos como bancos de dados orientados a grafos, grafos de conhecimento e redes complexas foram apresentados, elucidando como essas ferramentas podem ser úteis para modelar e analisar as interações acadêmicas que são alvo do presente trabalho. Além disso, foi introduzido o uso da linguagem *Cypher*, conceitos e técnicas de inteligência artificial que serão aplicados no contexto deste projeto. A compreensão desses conceitos permite estabelecer a base teórica necessária para o desenvolvimento e análise dos dados acadêmicos tratados no trabalho.

3 Trabalhos relacionados

Neste capítulo, é apresentado o método utilizado para realizar a busca de trabalhos relacionados, com o objetivo de encontrar os artigos similares mais relevantes para o desenvolvimento deste trabalho. Além disso, serão detalhados seis artigos, abrangendo desde a Seção 3.1 até a Seção 3.6, destacando seu impacto tanto positivo quanto negativo na presente pesquisa. Em seguida, na Seção 3.7, serão apresentadas as considerações finais deste capítulo.

Para a busca de trabalhos relacionados, foi utilizada a técnica de *snowballing*, conforme explicada por Wohlin (2014), o que permitiu embasar tecnicamente as ferramentas e abordagens escolhidas no desenvolvimento do trabalho. Através da busca realizada nas bases de dados científicas Google Scholar e Scopus, foi realizada uma coleta inicial de trabalhos que se assemelham à pesquisa.

Após a coleta inicial de artigos, foi aplicado um filtro com base na relevância dos títulos e resumos, excluindo trabalhos que não abordavam diretamente os temas de interesse ou que tinham enfoque em áreas muito distantes. Em seguida, foi realizada uma análise mais aprofundada do corpo dos artigos, considerando a qualidade das fontes, número de citações e a presença de implementações ou discussões sobre tecnologias e métodos similares aos empregados neste trabalho. Por fim, foram selecionados seis artigos que, além de apresentarem discussões relevantes, trouxeram perspectivas diferentes, permitindo uma comparação e uma base sólida para as escolhas metodológicas do presente trabalho.

3.1 Avaliação do desempenho docente

Fernandes (2008) define a avaliação docente como campo científico e uma prática social essencial para caracterizar, compreender, divulgar e melhorar uma ampla gama de questões que afetam as sociedades contemporâneas. O texto discute a avaliação do desempenho dos professores, destacando a importância de debater e refletir sobre questões relacionadas

a esse tema. O autor ressalta que é necessário um estudo aprofundado para compreender a substância e os fundamentos dessa avaliação, evitando cair em lugares comuns. Ele destaca que a avaliação do desempenho dos professores ainda não possui um conjunto estável de características mensuráveis que se apliquem a todos os contextos. No entanto, isso não impede o desenvolvimento de um sistema de avaliação que atenda aos interesses profissionais dos professores e às exigências de prestação pública de contas. O texto também menciona que estratégias de avaliação mais contextualizadas e específicas para cada professor são mais eficazes na melhoria do desempenho e competência docente do que sistemas uniformes aplicados hierarquicamente em todo o sistema educacional. Por fim, o autor destaca a importância de equilibrar uma perspectiva de desenvolvimento profissional contextualizada com uma perspectiva de responsabilização baseada em medidas de desempenho e eficácia.

O autor ainda apresenta uma lista de elementos fundamentais, como transparência e simplicidade, que devem ser considerados para uma avaliação docente eficaz. Essas informações serão utilizadas no presente trabalho para conduzir uma observação minuciosa e uma análise crítica do modelo proposto neste contexto, a fim de compará-lo com o modelo atual para verificar sua eficácia.

3.2 Semantic Data Integration for Knowledge Graph Construction at Query Time

O artigo menciona que com a evolução da *web* de documentos para uma *web* de serviços e dados, resultou em uma maior disponibilidade de dados em quase todos os domínios. Como houve essa mudança, várias fontes de dados diariamente publicam dados de diferentes maneiras e portanto, podem ter diferentes capacidades de pesquisa. Isso exige técnicas de integração de dados que forneçam uma visão unificada dos dados publicados.

É proposto por (COLLARANA et al., 2017) uma abordagem de integração semântica de dados chamada *FuhSen*, que utiliza capacidades de pesquisa por palavras-chave, estruturadas das fontes de dados da *web* que gera grafos de conhecimento sob demanda, mesclando dados coletados de fontes disponíveis na *web*. Os grafos de conheci-

mento resultantes modelam a semântica ou o significado dos dados mesclados em termos de entidades que satisfazem consultas por palavras-chave e os relacionamentos entre essas entidades. O *FuhSen* utiliza *Resource Description Framework* (RDF) para descrever semanticamente as entidades coletadas e medidas de similaridade semântica para decidir a relação entre as entidades que devem ser mescladas.

Os resultados da avaliação empírica sugerem que o *FuhSen* é capaz de integrar efetivamente informações dispersas em diferentes fontes de dados. Os experimentos indicam que a técnica de integração baseada em moléculas implementada no *FuhSen* integra dados em um grafo de conhecimento de forma mais precisa do que as técnicas de integração existentes. A abordagem de integração de moléculas RDF desenvolve um paradigma de integração novo, incorporando elementos de dados vinculados e mecanismos de pesquisa inovadores.

O presente trabalho necessita de realizar uma integração de dados de diferentes fontes por possuir informações com modelos heterogêneos, isto é, cada fonte de dados possui sua própria estrutura, portanto, o modelo integrado que será desenvolvido será melhor estruturado durante o processo de *data cleaning* proposto. Além disso, é notório que a forma de pesquisa por palavras-chave pode ser uma ferramenta poderosa para extrair conhecimento da base de dados, depois de populada. Por exemplo, pesquisando pelo nome do docente, o ano desejado e o eixo de ensino. Tais ferramentas podem facilitar o processo de retirada de métricas para a avaliação e análise da instituição.

3.3 PET-SQL: A Prompt-enhanced Two-stage Text-to-SQL Framework with Cross-consistency

O PET-SQL é um *framework* para a tradução de linguagem natural para SQL, que foi projetado para lidar com os desafios complexos de consultas baseadas em texto em grandes bancos de dados. Desenvolvido por Li et al. (2024), o projeto aborda dois principais problemas enfrentados pelos modelos atuais: a dificuldade em processar informações de esquemas extensos e a compreensão de intenções complexas dos usuários. Ele propõe uma abordagem em duas etapas para melhorar a geração de consultas SQL por LLMs, focando

em utilizar o aprimoramento de *prompts* e a verificação cruzada entre diferentes LLMs para aumentar a precisão e a consistência dos resultados.

Na primeira etapa, o *framework* apresenta uma representação de *prompt* aprimorada chamada representação referencial. Essa representação inclui informações sobre o esquema do banco de dados e amostras aleatórias de valores de células das tabelas, ajudando as LLMs a gerar uma consulta SQL preliminar (PreSQL). Para melhorar a precisão da geração, pares de perguntas e consultas SQL são recuperados como exemplos *few-shot* para guiar o modelo. Após gerar a consulta *PreSQL*, o *framework* realiza o processo de *schema linking*, que consegue simplificar as informações do esquema e extrai apenas os dados relevantes para a tarefa.

Na segunda etapa, é utilizado um processo chamado *cross-consistency*, que se baseia em usar múltiplos LLMs com a finalidade de verificar a consistência entre os resultados gerados, para tentar alcançar uma maior precisão. Como mencionado no artigo, essa abordagem é diferente da *self-consistency*, onde apenas um modelo revisa suas próprias respostas. O uso de diferentes modelos proporciona uma validação cruzada mais robusta, buscando eliminar ambiguidades e erros na consulta final.

Com essa estratégia, o PET-SQL atingiu resultados no *benchmark Spider*, com uma precisão de 87,6%, superando várias abordagens anteriores. O *framework* demonstra uma capacidade promissora de lidar com esquemas de bancos de dados complexos e melhorar significativamente a tradução de linguagem natural para SQL, tornando-o uma solução eficiente e escalável para aplicações do mundo real.

Como o presente trabalho pretende facilitar consultas ao banco de dados com o uso de linguagem natural, os conceitos e estratégias utilizados por (LI et al., 2024) podem ser úteis para o desenvolvimento do projeto.

3.4 Measuring higher education performance in Brazil

Este estudo está relacionado ao desempenho e a eficiência das instituições educacionais, incluindo o magistério superior, ao examinar o Instituto Federal de Educação, Ciência e

Tecnologia (IFET), que consiste em unidades educacionais em todo o Brasil, abrangendo vários níveis de ensino.

Wanke et al. (2021) constroem e analisam uma matriz de covariância composta por um grupo de medidas de eficiência e um grupo de indicadores de desempenho utilizados pelo MEC. Durante o estudo é realizada a aplicação da *Technique for Order Preference by Similarity to Ideal Solution* (TOPSIS), desenvolvida por Lai, Liu e Hwang (1994), que é um método de tomada de decisão multicritério que busca identificar a melhor alternativa entre um conjunto de opções. Assim, os pesos de cada variável são otimizados para capturar a direção da relação entre os dois conjuntos de medidas de eficiência.

Foi identificado que os indicadores de desempenho educacional usados pelo MEC não apresentam uma forte relação com a maioria das medidas de eficiência desenvolvidas neste estudo, indicando assim que as medidas de eficiência usadas por Wanke et al. (2021) superam o uso do MEC de vários indicadores como insumos no processo de produção educacional.

A pesquisa como um todo tem uma visão mais generalizada sobre a eficiência da instituição, analisando o desempenho dela em vários aspectos além da avaliação da atividade docente. São objetos de análise tanto os recursos humanos, quanto os recursos financeiros. Entretanto, é possível utilizar no presente trabalho, um método parecido para a avaliação dos resultados. Wanke et al. (2021) durante a análise da eficiência do IFET trabalham com um modelo que possui entradas que são responsáveis por produzir saídas específicas, conforme pode ser observado na Figura 3.1, que faz uma adaptação do modelo original. Algumas entradas utilizadas para análise no estudo são, orçamento, novas matrículas, professores e funcionários, e as respectivas saídas são alunos graduados e alunos de graduação.

A utilização do modelo de avaliação por entradas e saídas é uma estratégia eficiente para analisar os resultados, assim como a utilização de TOPSIS que é interessante para tomar decisões mais assertivas para avaliar o desempenho docente.

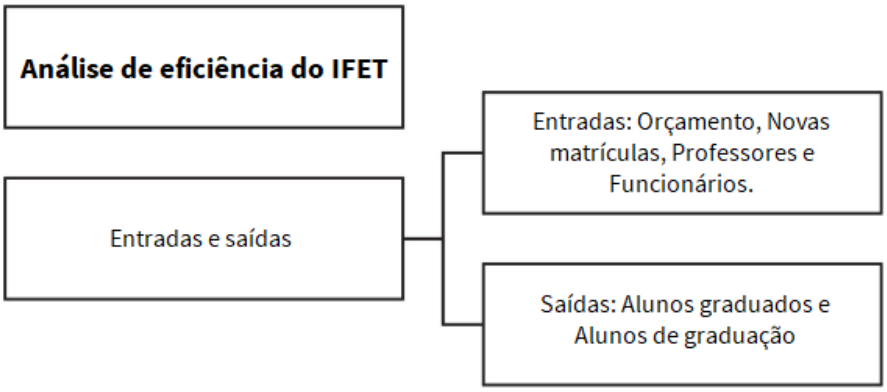

Figura 3.1: Exemplo adaptado do modelo de entradas e saídas baseado em: Wanke et al. (2021).

3.5 Query optimization techniques in graph databases

O artigo aborda o crescente interesse em utilizar bancos de dados de grafo e as técnicas de otimização de consultas. Esse tipo de banco surgiu como uma solução para superar as limitações dos bancos de dados tradicionais na armazenagem e gerenciamento de dados com estrutura de grafo (AMMAR, 2016). É mencionado que, atualmente, esses bancos de dados são requisitos essenciais para muitas aplicações que lidam com dados semelhantes a grafos, como redes sociais.

Com o aumento do uso no mercado, se fez necessário desenvolver e aplicar técnicas eficientes para trazer os dados desejados, com um desempenho satisfatório e melhor do que em bancos convencionais. A maioria das técnicas aplicadas para otimizar consultas em bancos de dados de grafo tem sido também utilizada em bancos de dados tradicionais, sistemas distribuídos ou são inspiradas na teoria dos grafos. No entanto, a reutilização em bancos de dados de grafo deve levar em consideração as principais características desses bancos, como a estrutura dinâmica, dados altamente interconectados e a capacidade de acessar eficientemente os relacionamentos dos dados.

Ammar (2016) descreve as principais técnicas utilizadas para otimizar as consultas em bancos de dados de grafo. Antes de passar pelas principais técnicas, o autor cita uma regra básica fundamental que é “cada nó do grafo contém um ponteiro direto

para seus elementos adjacentes e não são necessárias consultas de índice", o que faz a distinção aparente entre sistemas de bancos de dados de grafo e os tradicionais Sistema de Gerenciamento de Banco de Dados (SGBD). Portanto, todas as linguagens de consulta para bancos de dados de grafo são baseadas nessa regra, e os algoritmos estudados na teoria dos grafos constituem uma base sólida que as fundamenta.

Uma técnica discutida foi a *Pre-processing aggregate queries*, que busca otimizar o tempo de resposta e reduzir o custo de cálculos complexos. No contexto de consultas ego-cêntricas de agregação, onde um nó consome eventos de outros nós, foram propostas abordagens de transferência de eventos no momento da consulta ou pré-cálculo das respostas durante as atualizações nos nós produtores. A decisão de onde armazenar os dados materializados é baseada nos custos das tarefas de transferência e recuperação de eventos, permitindo o compartilhamento de agregações parciais entre diferentes consultas ego-cêntricas por meio de nós de agregação intermediários.

Com a revisão de técnicas de otimização de consultas em bancos de dados de grafo o presente trabalho se beneficia com possíveis estratégias para diminuir o tempo de resposta das consultas, tendo em vista que com o aumento do volume de dados contidos no banco, pelas informações de docentes do magistério superior, torna as consultas mais onerosas.

3.6 Sistema para Coleta e Avaliação de Relatórios Individuais de Trabalho

O trabalho de Carvalho (2022) está relacionado com a gestão pública e acadêmica, voltado a colaborar com a avaliação dos RITs através de uma plataforma online.

Carvalho (2022) desenvolveu um software, com o objetivo de gerar uma aplicação *web* para realizar a importação através de formulários, das planilhas do DCC e de arquivos baixados manualmente da plataforma *Lattes*. De posse dos dados, é feita uma análise através de diversos gráficos e relatórios individuais de trabalho de professores. Os dados são compilados em gráficos, a fim de tornar compreensível a sua descrição e facilitar a avaliação do trabalho desenvolvido pelos professores ao longo do trabalho acadêmico.

Com posse de todas essas informações, é possível ter uma visão geral de quais foram as atividades mais desenvolvidas, no ano corrente ou ao longo dos anos, quais eixos ou categorias possuem maior esforço, e como o professor se posiciona dentro do departamento e dentro da instituição.

Foi feita uma modelagem dos dados e os mesmos foram organizados por categorias, eixos e anos, respectivamente. Dessa forma, a evolução das atividades produzidas no departamento puderam ser detalhadas ao longo dos anos nos quais a coleta e importação foram realizados.

Os gráficos desenvolvidos no sistema, envolvendo dados de diversos RITs, fornecem base para o departamento tomar decisões pautadas na evolução dos dados, e não só no RIT anual como é feito atualmente. Na aplicação, é permitida a visualização das estatísticas dos departamentos, transparecendo a evolução da produção em uma visão global, assim como estatísticas da instituição, que é possível visualizar o gráfico da quantidade de atividades por categoria e também um outro gráfico permitindo representar a quantidade de atividades por eixos. Os eixos são divididos em ensino, pesquisa, extensão e gestão acadêmica.

A importação dos dados oriundos do *Lattes*, por exemplo, se restringiu aos artigos no ano de interesse. Entretanto, o currículo *Lattes* é a principal fonte de produção no Brasil, logo, existe diversas outras informações que podem ser extraídas, como projetos, orientações e participações em bancas.

O presente trabalho pretende preencher essa lacuna, automatizando a coleta dos arquivos oriundos da plataforma *Lattes*, para ter uma maior agilidade e também utilizar outras informações dos arquivos extraídos da plataforma, além de artigos de um determinado ano de interesse.

3.7 Considerações Finais

A Tabela 3.1 faz um comparativo entre os seis trabalhos relacionados que foram discutidos nas seções anteriores, de forma a considerar a abordagem dos temas e a utilização de técnicas que poderão ser utilizadas no presente Trabalho de Conclusão de Curso.

Nos trabalhos relacionados, constam estudos que utilizam técnicas que serão

Tabela 3.1: Tabela comparativa entre os trabalhos relacionados.

Nome do Trabalho	**Magistério no Ensino Superior**	**Obtenção de Dados**	**Inteligência Artificial**	**Banco de Grafos**	**Métricas para Avaliação Docente**
Fernandes (2008)	X				X
Collarana et al. (2017)		X		X	
Li et al. (2024)			X	X	
Wanke et al. (2021)	X				X
Ammar (2016)				X	
Carvalho (2022)	X	X			X

úteis para o desenvolvimento do presente trabalho. O artigo de Fernandes (2008), é útil para extrair informações relevantes para criar métricas que são capazes de julgar a evolução das instituições de ensino. Já o artigo de Collarana et al. (2017) utiliza uma estratégia que permite agrupar dados que serão extraídos de mais de uma fonte, de maneira que não gere conflitos de modelo, permitindo assim, o uso de informações mais precisas. O texto de Li et al. (2024) é enriquecedor no contexto de IA, pois demonstra um experimento que utiliza LLMs, RAG e a engenharia de prompt, que serão utilizadas no desenvolvimento do *RAG Chat*. Com um banco de dados já populado, Ammar (2016) esclarece como tornar as consultas ao banco de dados de grafos mais eficientes, mesmo em ocasiões em que existe um grande volume de dados. Por fim, Carvalho (2022) mostra a implementação de uma plataforma que busca facilitar a gestão do ensino público mostrando aos usuários, gráficos que expõem com clareza o progresso do departamento ao longo do tempo.

4 Desenvolvimento

Neste capítulo é apresentado o desenvolvimento do projeto, detalhando cada etapa, informando quais foram as ferramentas escolhidas e o motivo da escolha, passando por todo o processo de implementação, desde a importação dos dados das planilhas RIT até as análises realizadas sobre o modelo de grafo construído.

4.1 Visão Geral

A primeira parte do desenvolvimento do projeto, está representada na Figura 4.1. Inicialmente, o módulo chamado de *RIT Importer* realiza a importação dos dados das planilhas de lançamento do RIT para o departamento de Ciência da Computação da UFJF. Ele extrai as informações das atividades docentes e estrutura um arquivo JSON separado por professor, contendo todas as atividades associadas a ele. Para a modelagem do grafo, foi desenvolvido um programa em *JavaScript* que itera sobre a pasta onde estão armazenados os arquivos JSON. Durante essas iterações, são geradas consultas *cypher* para popular o banco de dados Neo4j, constituindo o grafo modelado e possibilitando consultas e análises subsequentes.

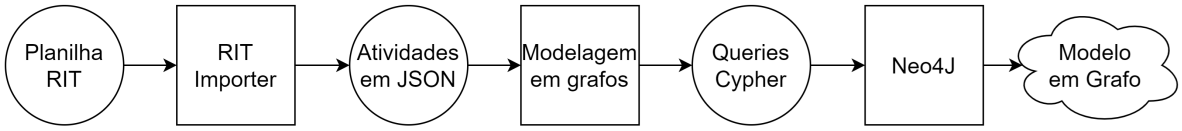

Figura 4.1: Diagrama da primeira etapa do trabalho, que consiste na importação dos dados da planilha RIT com o *RIT Importer*, que gera um arquivo JSON que por sua vez é utilizado na modelagem em grafos para criar as consultas na linguagem *Cypher* que vão popular o Neo4j, resultando no modelo em grafo.

Posteriormente, na segunda etapa representada na Figura 4.2, foram identificados dados incorretos, como erros de preenchimento nos nomes de alunos e outras inconsistências. A importação desses dados com lançamento incorreto afetou o modelo, gerando ligações e nós imprecisos. Para resolver essa questão, foi criado um sistema de

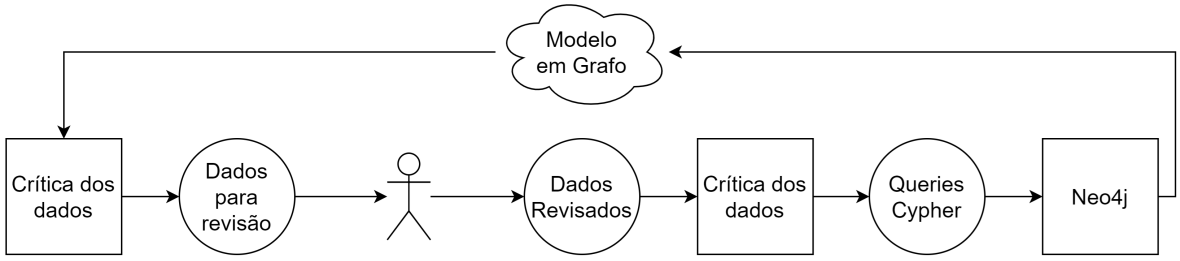

Figura 4.2: Diagrama da segunda etapa do trabalho, que ilustra o sistema de crítica de dados, que a partir do modelo em grafo, traz para o usuário dados que são suspeitos de inconsistências, para realizar a revisão. Com a ação do usuário o sistema gera consultas *Cypher* que são executadas no banco de dados e alteram o modelo em grafo.

crítica de dados, que, a partir do modelo contaminado com erros de digitação e inconsistências, identifica possíveis problemas. O sistema permite que o usuário revise esses dados e, após a validação ou correção dos erros gera as consultas em *cypher* de correção, resultando em um modelo mais preciso.

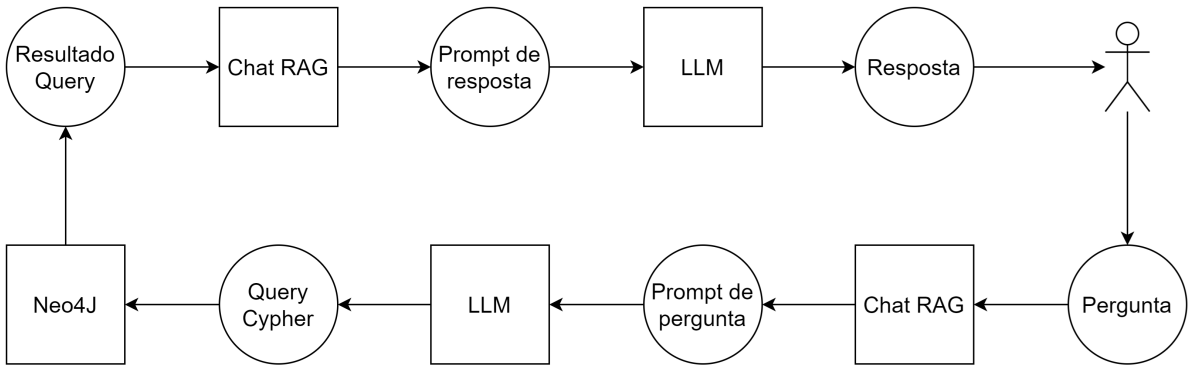

Figura 4.3: Diagrama da terceira etapa do trabalho, que ilustra a interação do usuário com o *Chat RAG* através de uma pergunta em texto com linguagem natural, que o sistema utiliza somando a um *Prompt* de pergunta, que é levado a um LLM, que gera consultas *Cypher* para serem executadas no Neo4j. Com o resultado da consulta o *Chat RAG* soma a um *Prompt* de resposta e o envia para o LLM gerar uma resposta mais clara para o usuário.

A terceira etapa representada na Figura 4.3, visa oferecer uma maneira descomplicada e flexível de obter respostas sobre o modelo. Foi desenvolvida uma interface na forma de *chat* que utiliza as técnicas de RAG, permitindo que uma LLM interaja diretamente com o grafo e forneça respostas contextualizadas.

Por fim, na Figura 4.4 o projeto inclui uma análise de redes complexas, que possibilita a detecção de comunidades dentro do grafo, permitindo a extração de novas

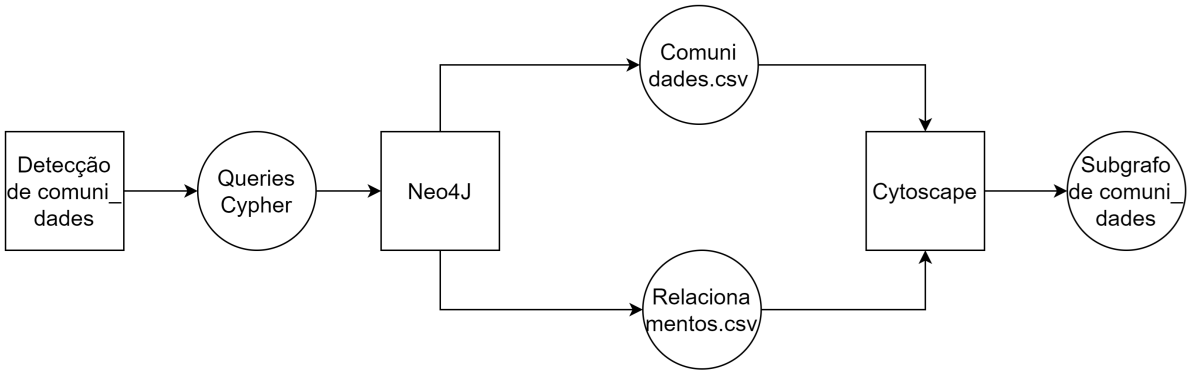

Figura 4.4: Diagrama da quarta etapa do trabalho, que ilustra o processo de análise e detecção de comunidades, que a partir de consultas *Cypher* com suporte do plugin GDS, gera dois arquivos CSV, um com os docentes e comunidades e outro com os relacionamentos dos docentes, que ao serem importados para a ferramenta *Cytoscape*, exibe o subgrafo de comunidades entre os docentes.

informações a partir dessas estruturas, com o auxílio da ferramenta *Cytoscape*.

4.2 Origem de dados

A orgiem de dados utilizada neste trabalho são as planilhas de coleta dos RIT dentro do DCC da UFJF. Essas planilhas são preenchidas pelos docentes para relatar as atividades realizadas ao longo do ano letivo. Essas atividades incluem aulas ministradas, funções administrativas desempenhadas, cargos ocupados, publicações acadêmicas, entre outras. A planilha é estruturada com uma página dedicada a cada docente e é organizada em cinco eixos: Atividades Administrativas, Atividades de Ensino, Atividades de Pesquisa e Extensão, Afastamento e Capacitação, e Outras Informações. Os docentes preenchem a planilha conforme o exemplo ilustrado na Figura 4.5.

No processo de importação, a integração entre o programa *JavaScript* e as planilhas é realizada através da biblioteca *GoogleSpreadsheet*, que facilita o acesso direto aos dados armazenados no Google Sheets. Uma vez estabelecida a conexão, os dados são carregados em memória, e o processo de importação é iniciado. Esse processo é gerido por uma máquina de estados que percorre sistematicamente as células da planilha, extraindo as informações sobre as atividades relatadas.

O resultado desse processamento é uma pasta com um arquivo JSON gerado para cada docente, contendo informações detalhadas como o Sistema Integrado de Ad-

Atividades de Ensino

Preencha os campos abaixo listando as disciplinas lecionadas no ano letivo.

Período	Disciplina (código, nome e turma)	Núm Alunos	Carga Horária
1	ABC123 - NOME DISCIPLPINA I	10	60
1	ABC123 - NOME DISCIPLPINA II	20	60
1	ABC123 - NOME DISCIPLPINA III	30	30
3	ABC123 - NOME DISCIPLPINA IV	40	30
1	ABC123 - NOME DISCIPLPINA V	50	60

Figura 4.5: Ilustração de parte da planilha RIT, da maneira que os docentes preenchem.

ministração de Recursos Humanos (SIAPE), o nome do docente, o ano de referência, e todas as atividades realizadas, que por sua vez, estão devidamente organizadas por eixo, tipo de atividade e categoria. Além disso, o arquivo gerado possui metadados sobre o estado de processamento, para garantir que os dados daquele docente foram completamente extraídos durante do processo de importação.


```
{
  "state": "PROCESSAMENTO COMPLETO",
  "SIAPE": "1234567",
  "nome": "DOCENTE EXEMPLO",
  "ano": 2024,
  "ensino": [
    {
      "natureza": "Mestrado",
      "aluno": "Aluno da Silva",
      "categoria": "Participações em bancas"
    }
  ],
  "pesquisaEExtensao": [
    {
      "projeto": "Nome do Projeto",
      "orgao": "Orgão do Projeto",
      "qtd": 12,
      "unidade": "meses",
      "tipo": "Projeto de Pesquisa com fomento",
      "categoria": "Coordenação de Projeto de Pesquisa"
    },
  ],
  "afastamentoECapacitacao": [],
  "outras": []
}
```


Figura 4.6: Exemplo de um arquivo JSON, contendo as informações do docente e suas atividades, separadas por eixo.

4.2.1 Modelo em grafos

Com os dados das atividades docentes organizados, o próximo passo no desenvolvimento deste trabalho foi a modelagem dos dados no BDOG Neo4j[1]. Foi utilizada a linguagem *Cypher* para realizar as consultas e operações no banco. Foi adicionado ao Neo4j, o *plugin Awesome Procedures On Cypher* (APOC)[2], que provê várias *procedures* e funções, agregando valor com algoritmos já implementados. Com o ambiente preparado, o processo de modelagem foi iniciado com cautela, para que a estruturação dos nós e seus relacionamentos tenham um impacto positivo nas consultas que serão realizadas.

Para a modelagem do grafo foi formulado de um conjunto de perguntas que o grafo deveria ser capaz de responder. Essa abordagem orientada a questionamentos permitiu identificar as entidades e os relacionamentos necessários para a estrutura do grafo. Por exemplo, a pergunta "Quais aulas o docente X ministrou?" revelou a necessidade de criar as entidades "Docente" e "Aula", além de estabelecer um relacionamento "MINISTRA". Esse relacionamento é ilustrado na Figura 4.7.

Figura 4.7: Um professor (círculo roxo, à esquerda) é ligado a uma disciplina (círculo laranja, à direita) por um arco de flecha que representa que ele ministrou a disciplina.

O modelo de grafo desenvolvido está ilustrado na Figura 4.8. Cada nó do grafo representa uma entidade relevante, como Docente, Atividade, Categoria, entre outros, enquanto as arestas indicam as relações entre essas entidades, refletindo a estrutura, as interações presentes nas planilhas de RIT e também, ainda no presente trabalho, será possível apontar outras interações entre as entidades.

O modelo do grafo passou por diversas revisões ao longo do desenvolvimento do projeto. Essas modificações foram necessárias para atender melhor as consultas e permitir uma exploração mais detalhada das informações, conforme novas demandas foram surgindo. Esse processo de ajuste contínuo é comum na modelagem de grafos, onde a

[1]Banco de dados Neo4j Desktop 5.12.0. Disponível em ⟨https://neo4j.com/download/⟩.
[2]APOC 5.12.0. Disponível em ⟨https://neo4j.com/labs/apoc/⟩.

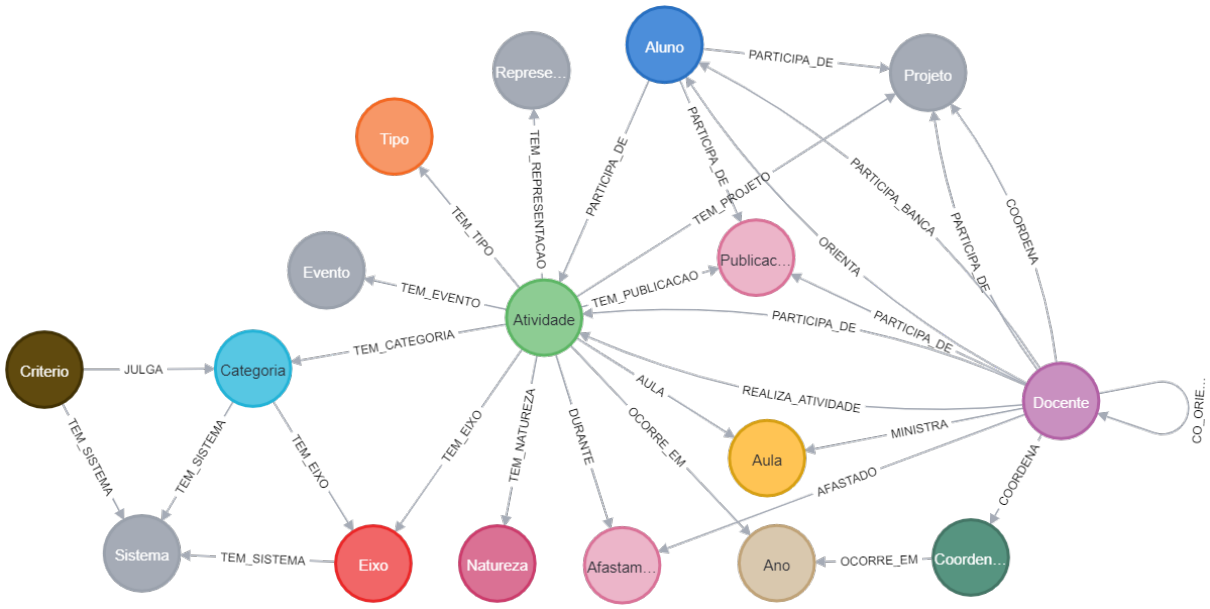

Figura 4.8: Esquema extraído do Neo4j, representando os nós e seus possíveis relacionamentos.

evolução do esquema está frequentemente associada à descoberta de novas relações e à necessidade de melhorar as operações de consulta .

Inicialmente, a principal ligação no grafo era baseada em nós representando docentes realizando uma atividade, a qual, por sua vez, possuía uma categoria. Essa estrutura é mostrada na Figura 4.9.

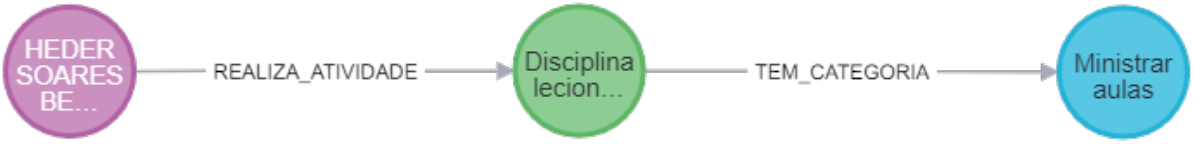

Figura 4.9: Ligação dos principais nós, Docente, Atividade e Categoria, representando a realização de uma atividade por um docente.

Com o objetivo de tornar o grafo mais completo e facilitar certas consultas, foram adicionados outros nós e relacionamentos, como ilustrado na Figura 4.10. Essa expansão do modelo proporcionou um nível mais detalhado de consultas e análises, permitindo uma caracterização mais individual das atividades e a criação de relacionamentos específicos conforme o contexto. Por exemplo, ao adicionar o nó “Aula”, foi possível atribuir propriedades detalhadas, como o número de alunos, carga horária e período da disciplina, entre outras. Essa individualização facilitou a extração dessas informações, tor-

nando as consultas com uma granularidade mais fina e, especialmente em consultas mais especificas que serão possíveis de ser respondidas com o sistema de chat, que é explicado na Seção 4.3.

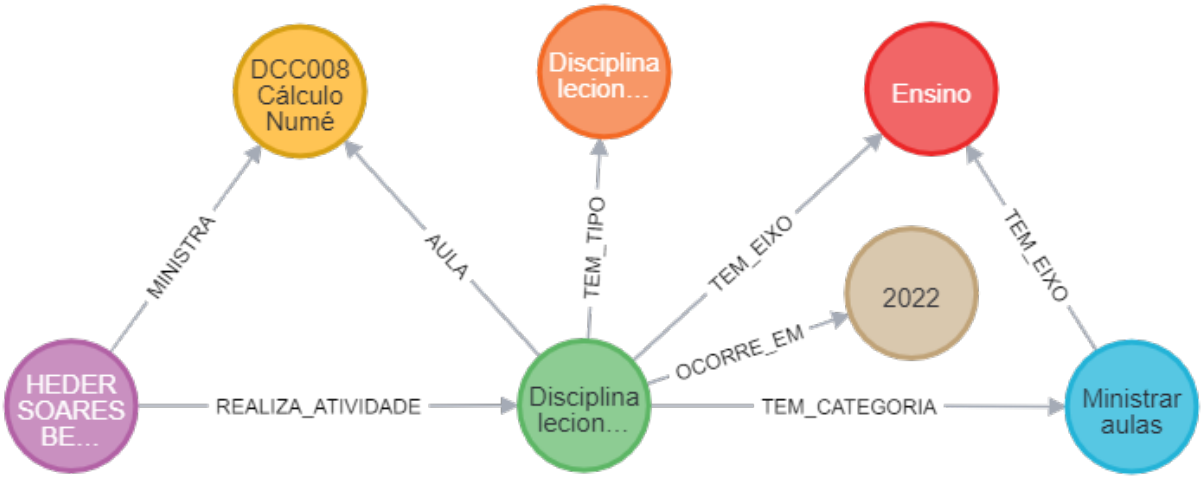

Figura 4.10: Expansão do grafo com novos nós e relacionamentos, já tendo criado nós mais específicos a partir da atividade realizada, como o nó Aula em amarelo.

Antes de começar a inserir os dados no banco, é feita a definição de *constraints*, que asseguram a unicidade de determinados nós e garantem que campos essenciais sejam obrigatórios. Essa abordagem evita duplicações e assegura que as informações obrigatórias estejam presentes.

O processo de construção do grafo é realizado iterativamente, onde cada arquivo JSON, com informações de um RIT, é carregado na memória e processado. Como as atividades podem ter diferentes propriedades no JSON, conforme ilustra a Figura 4.6, foi desenvolvido uma função que leva em consideração cada propriedade da atividade que está analisando e mapeia o objeto recebido, criando uma consulta *cypher* para realizar a inserção dos dados no *Neo4j*. Esse procedimento é repetido para todos os arquivos, resultando em um grafo populado com os dados das atividades realizadas pelos docentes.

4.3 Chat com geração aumentada via recuperação

Com a intenção de flexibilizar o acesso às informações que constam no banco de grafos, foi desenvolvido um sistema de chat que utiliza uma variação da técnica de inteligência artificial RAG. Dando liberdade ao usuário para fazer perguntas em linguagem natural

para o modelo, recuperando informações específicas do banco de dados Neo4j, e com isso, gerando respostas em linguagem natural com base nas informações recuperadas do contexto de foco, sem precisar treinar novamente a LLM.

Para a integração entre os dados do Neo4j e a LLM, foi utilizada a classe ChatOpenAI da biblioteca LangChain[3], que permite a utilização dos modelos de linguagem da OpenAI. Na instanciação dessa classe, foram configurados parâmetros como o modelo de linguagem, temperatura e número máximo de tentativas. Foram experimentados os modelos gpt-3.5-turbo-0125, gpt-4o-mini e gpt-4o que oferecem suporte para fine-tuning. Durante a experimentação, observou-se que os modelos mais simples cometiam mais erros de sintaxe na geração das consultas Cypher. E como esperado, o modelo mais avançado, gpt-4o, se mostrou mais sólido, errando menos nas criações das consultas, portanto, foi o escolhido para seguir com o desenvolvimento do projeto.

Com a intenção de diminuir a possibilidade de alucinações da LLM, a temperatura do modelo de linguagem foi ajustada para zero, já que com valores maiores do que zero, as consultas foram criadas com maior chance de erros, como o uso de relacionamentos que não existiam no grafo, nomes de propriedades que também não constavam no modelo do banco ou então fazendo a utilização de palavras-chave usadas em bancos relacionais.

Já o número máximo de tentativas foi limitado para dois, para evitar o consumo excessivo de recursos da plataforma e também para fazer com que o sistema tente novamente antes de retornar um erro ao usuário, em caso de eventuais falhas na comunicação com a API.

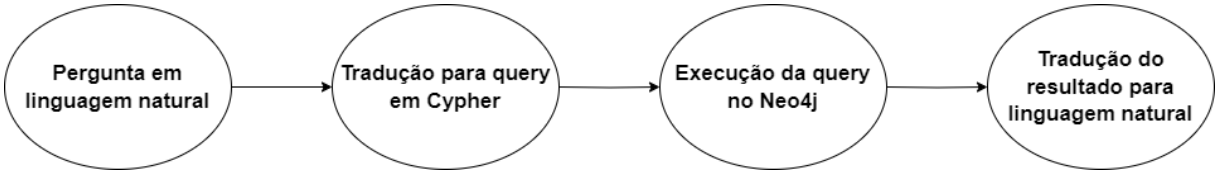

Figura 4.11: Fluxo do funcionamento do chat, partindo da pergunta realizada em linguagem natural, passando para a tradução dessa pergunta para uma consulta em *Cypher*, executando no banco de dados e traduzindo o resultado novamente para linguagem natural.

O fluxo de funcionamento do chat é descrito na Figura 4.11. O processo

[3]Biblioteca em python para uso com LLM. Disponível em ⟨https://python.langchain.com/⟩.

começa com a formulação da pergunta em linguagem natural pelo usuário, que é então traduzida em uma query Cypher pela LLM, utilizando um *prompt* específico. A consulta é executada no banco de dados Neo4j, e o resultado é posteriormente traduzido de volta para uma resposta em linguagem natural, utilizando outro prompt desenhado para formatar e apresentar a resposta de forma compreensível ao usuário.

Durante o desenvolvimento, aplicaram-se conceitos de engenharia de prompt para criar e refinar as instruções passadas à LLM. Um prompt inicial foi desenvolvido para traduzir perguntas em linguagem natural para consultas Cypher. Esse prompt inclui uma instrução para a caracterização do modelo: *"Você é um tradutor especialista em Neo4j e Cypher que converte português brasileiro para Cypher com base no esquema Neo4j fornecido, seguindo as instruções abaixo:"...* Também foram adicionadas instruções específicas para evitar erros comuns, como o uso de informações que não estão no grafo. Por exemplo, a instrução *"Use apenas nós e relacionamentos que estão presentes no schema"* foi incluída após a observação dessa classe de problemas nas consultas geradas. O esquema do grafo foi passado como parâmetro, e uma lista de exemplos de perguntas e suas respectivas respostas em *cypher* esperadas para guiar o modelo em consultas mais complexas. Finalmente, a pergunta do usuário é inserida no prompt para a geração da query.

Após a criação e execução da query no Neo4j, foi desenvolvido um segundo prompt para a tradução do resultado de volta para linguagem natural. Esse prompt inclui instruções como *"Você é um assistente que ajuda a formar respostas agradáveis e compreensíveis para humanos"*. A versão final dos prompts foi alcançada após uma série de testes e refinamentos, resultando em uma configuração capaz de gerar respostas satisfatórias para o projeto.

Conforme a Figura 4.12, é possível ver a divisão do chat em três blocos que contêm, respectivamente: o chat com as perguntas e respostas, a query que foi gerada pela llm e os resultados retornados pelo banco de dados.

4.4 Sistema de avaliação de atividades docentes

Uma demanda comum do magistério superior é a necessidade de avaliar os docentes pelas atividades exercidas. Por vezes, o mesmo conjunto de atividades precisa ser avaliado

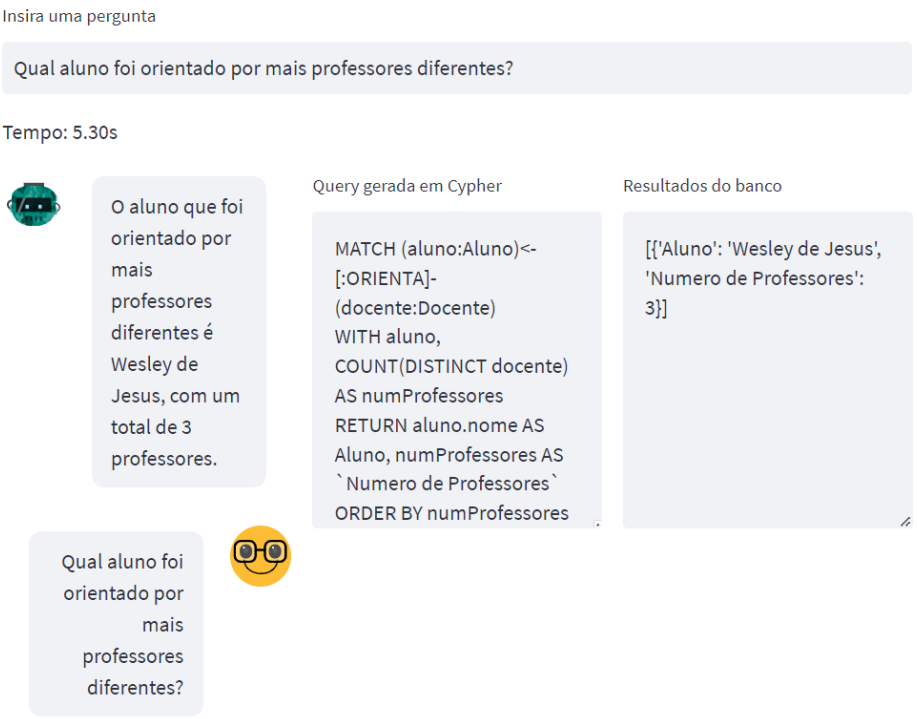

Figura 4.12: Exemplo de utilização do chat

sob diferentes perspectivas e critérios. Para atender a essa necessidade, foi desenvolvido um sistema flexível de contabilização de pontos, que permite a avaliação das atividades docentes segundo múltiplos critérios, adaptando-se às diversas exigências institucionais. A Figura 4.13 mostra parte do grafo referente a avaliação das atividades.

Cada atividade realizada pelos docentes é categorizada, e os nós que representam os critérios de avaliação mantêm o relacionamento *JULGA* com os nós de categoria. Esse relacionamento contém um peso associado, que determina a quantidade de pontos serão atribuídos a uma atividade dentro de uma determinada categoria.

Por meio da interface do chat, como mostra na Figura 4.14 o sistema oferece a possibilidade de realizar a contagem de pontos. O usuário pode perguntar diretamente: "Quantos pontos o docente X obteve no ano Y, segundo o critério Z?". É interessante que a pesquisa possa ser realizada por linguagem natural via chat, pois possibilita que o usuário faça uma busca mais específica. Com a modelagem do grafo possibilitando mais de um critério de avaliação, resultou em um sistema mais flexível para a avaliação de atividades.

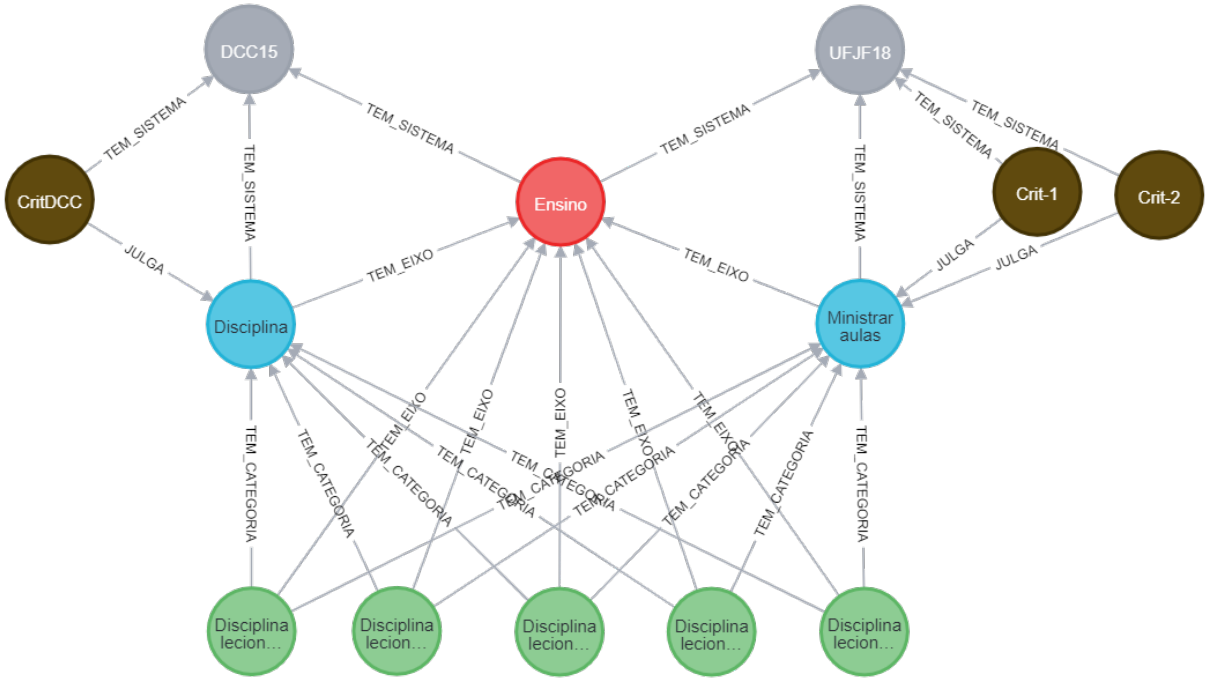

Figura 4.13: Parte do grafo mostrando os nós e relacionamentos de sistema, critérios , atividades e categorias.

4.5 Crítica dos dados

Dado que os docentes da UFJF são responsáveis por preencher manualmente a planilha de RIT, existe a chance de ocorrerem erros de digitação. Por exemplo, quando diferentes docentes citam o mesmo projeto, mas o digitam de forma ligeiramente diferente, criam-se dados equivocados no modelo. Essa inconsistência na entrada dos dados leva à criação de múltiplos nós de projetos no banco de dados Neo4j, quando, na realidade, deveria haver apenas um. Situações semelhantes ocorrem com os nomes dos alunos, resultando na duplicação de entidades que deveriam ser únicas.

Para mitigar essas inconsistências, foi implementado um processo de crítica dos dados, focado principalmente nos problemas de duplicidade em projetos e alunos, identificados como os mais frequentes. A solução desenvolvida envolve a colaboração do usuário na revisão e correção desses casos suspeitos de erro. Na aplicação web, os usuários têm acesso a uma tela de revisões que apresenta possíveis duplicidades em projetos e alunos.

A detecção dessas inconsistências é realizada utilizando o algoritmo de comparação de strings baseado na distância de Jaro-Winkler. Uma distância máxima foi

Figura 4.14: Interação com o chat, através de pergunta com linguagem natural a fim de saber quantos pontos um determinado docente tirou no ano de 2022.

empiricamente definida para identificar strings que, apesar de diferentes, são muito semelhantes, como variações mínimas nos nomes de projetos ou alunos.

Uma vez que o sistema identifica um possível caso de duplicidade, o sistema traz uma lista das ocorrências como ilustra a Figura 4.15. O usuário pode revisar e decidir qual entidade deve ser mantida como correta. Com a ação do usuário, o sistema executa a fusão das entidades duplicadas, transferindo todos os relacionamentos do nó que será excluído para o nó correto. Esse processo colabora com a integridade dos dados no banco e permite que as correções sejam refletidas em toda a estrutura de relacionamento, preservando a coerência das análises subsequentes. Observou-se que, após essas correções, características de redes complexas, como a formação de comunidades, foram impactadas.

Foi observado que, em alguns casos, o campo destinado ao nome dos alunos continha caracteres especiais ou informações inadequadas, como o nome de projetos ou

Resultados da Revisão de Alunos Duplicados:

IdA	NomeA	IdB	NomeB	DistanciaJaroWinkler	Ação
775	Karla Gabriele Florentino da Silva.	4782	Karla Gabriele Florentino da Silva	0.005714	Manter A Manter B
93	Bryan Carolino Muiz Barbosa	5702	Bryan Carolino Muniz Barbosa	0.007143	Manter A Manter B
1352	Gabriel Henrique de Souza.	2150	Gabriel Henrique de Souza	0.007692	Manter A Manter B
4482	RAFAEL BRAGA LADIERA DUTRA	5286	RAFAEL BRAGA LADEIRA DUTRA	0.007692	Manter A Manter B
1515	Nathan Manera Magalhães	4234	Nathan Manera Magalhães.	0.008333	Manter A Manter B
778	Aleksander Yacovenco.	5348	Aleksander Yacovenco	0.009524	Manter A Manter B
461	João Vitor Oliveira	735	João Victor Oliveira	0.010000	Manter A Manter B
1057	Lívia Pereira Ozório	3481	Lívia Pereira Ozório	0.012698	Manter A Manter B
3208	Matheus Avila Moreira de Paula	3739	Matheus Ávila Moreira de Paula	0.013333	Manter A Manter B
4538	Matheus Farjado Galvão	5577	Matheus Fajardo Galvão	0.013636	Manter A Manter B
3363	YAGHO MATTOS DA ROCHA - PIBIC	3366	YAGHO MATTOS DA ROCHA - PIBIT	0.013793	Manter A Manter B

Figura 4.15: Exibição do sistema de crítica dos dados mostrando uma lista com os Nós possivelmente duplicados para correção.

símbolos incomuns. Dado que esse campo deve conter exclusivamente o nome do aluno, foi desenvolvida uma função para identificar automaticamente essas irregularidades, verificando a presença de caracteres não usuais e tamanhos de strings fora do padrão. Esse mecanismo, integrado ao sistema, permite ao usuário corrigir diretamente esses dados, editando o nome do aluno na tabela, como ilustrado na Figura 4.16.

4.6 Detecção de Comunidades

O banco de dados Neo4j permite a instalação de *plugins* que ampliam suas funcionalidades. O GDS [4] é um *plugin* responsável por fornecer implementações de algoritmos de análise de grafos, como o algoritmo de Louvain.

Neste projeto, o algoritmo de Louvain foi adotado para identificar comunidades entre docentes, considerando os nós "Docente" e o relacionamento "CO_ORIENTA". Esse relacionamento foi definido quando dois ou mais docentes orientam o mesmo aluno, independentemente de se a orientação ocorreu no mesmo projeto ou atividade. Por exemplo, se o professor X orientou um aluno em uma atividade e o professor Y orientou o mesmo aluno em outra, ambos terão o relacionamento "CO_ORIENTA", como mostrado na Figura 4.17.

[4] O plugin GDS versão 2.6.8. Disponível em ⟨https://neo4j.com/docs/graph-data-science/2.6/⟩.

Resultados da Revisão de Nomes Incorretos:

	Nome	IdNo
0	16 bancas, 12 nas Ciências Exatas	180
1	2 Qualificação de Mestrado UDESC, Doutorado UFPE	182
2	5 defesas de mestrado UFF	185
3	5 defesas de doutorado no PPGEGC	188
4	ALLAN AMARAL SANT´ANNA ROCHA	263
5	Gabriel Bronte (Projeto - Pró-Inclusão 3a edição)	365
6	Wellington Pereira (Projeto - Encontro)	367
7	Daniel Borges de Oliveira (Projeto - Encontro)	380
8	Thomás Sousa Causin Alves (Projeto - Encontro)	382
9	Julia Araújo (Projeto - Encontro)	384
10	Marina Araújo (Projeto - Encontro)	386

Salvar Alterações

Figura 4.16: Lista com nomes de alunos possivelmente incorretos, para edição.

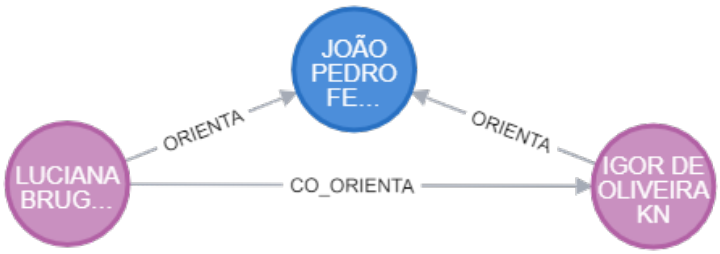

Figura 4.17: Exemplo de dois docentes que orientam o mesmo aluno, resultando no relacionamento CO_ORIENTA entre os docentes.

Na Figura 4.17, Luciana orienta João Pedro em um treinamento profissional, enquanto Igor o orienta em um projeto do GET. Para o presente projeto, esse cenário justifica a criação do relacionamento "CO_ORIENTA" entre os dois docentes.

O processo de detecção de comunidades foi realizado em várias etapas. Inicialmente, uma projeção de um subgrafo foi criada em memória, com base nos nós "Docente"e nos relacionamentos "CO_ORIENTA". Nessa projeção, os relacionamentos foram tratados como não direcionados, de modo que a direção original não fosse considerada nas análises subsequentes. Em seguida, o algoritmo de Louvain foi aplicado para detectar as comunidades, retornando os docentes e os respectivos identificadores das comunidades.

Os resultados dessa execução foram armazenados em um arquivo CSV denominado “Comunidades.csv”. Além disso, foi gerado um arquivo contendo as conexões entre docentes, com as colunas “source” e “target”, representando os nós de origem e destino, respectivamente, em um arquivo intitulado “Relacionamentos.csv”. Esses dados foram então importados no software Cytoscape [5] , que é uma ferramenta que foi originalmente desenhada para trabalhar com pesquisas biológicas, mas atualmente é uma plataforma com diversas funções para a análise de redes complexas (TEAM, 2024). Com as informações no Cytoscape, foi utilizado o layout “Group Attributes Layout”foi utilizado para agrupar os nós com base no atributo *communityId*, permitindo a visualização clara das comunidades detectadas, como mostrado na Figura 4.18.

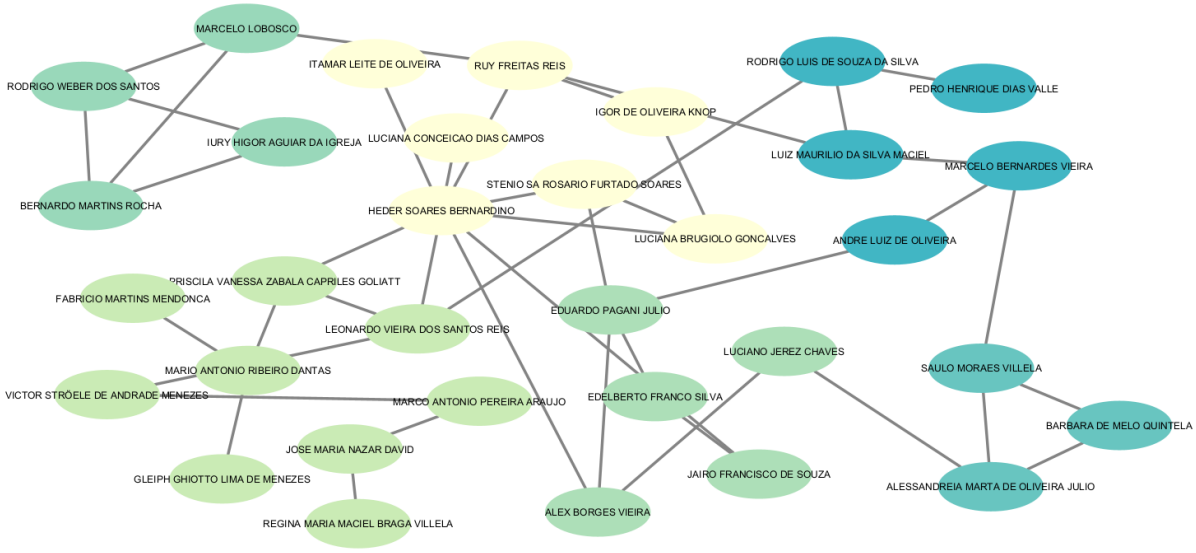

Figura 4.18: Comunidades detectadas na análise inicial, antes da crítica dos dados.

Fizemos duas execuções de detecção de comunidades, a primeira se deu com o banco de dados populado com informações extraídas diretamente do *RIT Importer*, sem a aplicação prévia de uma crítica detalhada dos dados, que está representada na Figura 4.18.

Reconhecendo a importância da qualidade dos dados, procedeu-se com a etapa de crítica e refinamento das informações. Com o apoio do processo já mencionado na Seção 4.5, parte dos nós e relacionamentos inconsistentes ou incorretos foram corrigidos. Com o grafo ajustado, uma nova execução do algoritmo de Louvain foi realizada e o processo para a exibição das comunidades foi repetido, resultando na Figura 4.19.

[5]Cytoscape 3.10.2. Disponível em ⟨https://cytoscape.org/download.html⟩.

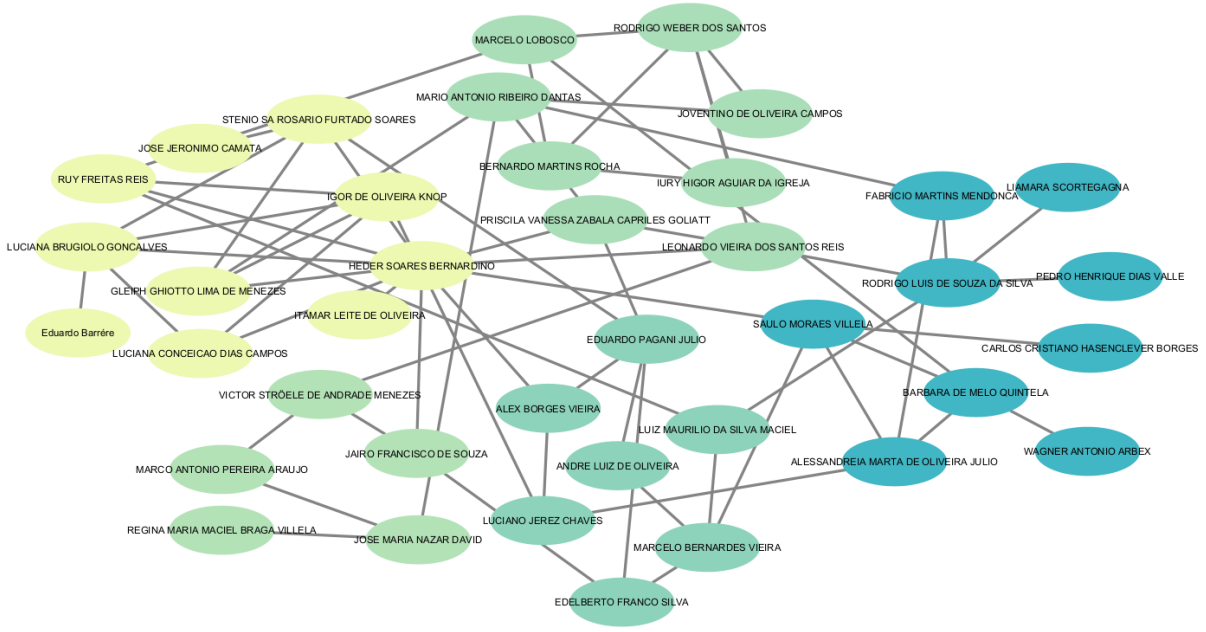

Figura 4.19: Comunidades detectadas após a crítica dos dados.

Os resultados da segunda execução evidenciaram que a crítica dos dados provocou mudanças na detecção das comunidades. Notou-se mudanças relevantes nas comunidades identificadas, o número de comunidades passou de seis para cinco, e alguns docentes que anteriormente não apareciam no grafo, devido à ausência de relacionamentos de co-orientação, passaram a ser incluídos. Essas alterações destacam a importância de garantir a qualidade e precisão dos dados no modelo de grafo, evidenciando como informações corretas influenciam diretamente a identificação de comunidades e a compreensão das interações entre docentes.

4.7 Interface Web

Durante o processo de escolha das linguagens e ferramentas para o desenvolvimento da interface, a simplicidade e a agilidade foram os critérios centrais. A linguagem *Python*, na versão 3.11.5, foi selecionada devido à sua flexibilidade e a variedade de bibliotecas que facilitam a construção rápida de aplicações web. A biblioteca *Streamlit*, versão 1.21.1, foi escolhida para a criação da interface web, por oferecer uma maneira prática e rápida de desenvolver interfaces interativas, sem a necessidade de lidar com configurações complexas de *frontend*.

A interface foi dividida em duas partes principais: o chat, responsável por responder perguntas dos usuários e interagir com o banco de grafos, e a seção de revisão de dados, que permite a correção de imprecisões nos dados.

O módulo `chat.py` implementa um chat utilizando o *Streamlit* para fornecer uma interface simples, na qual os usuários podem fazer perguntas relacionadas às atividades docentes e obter as respostas. A interface, como pode ser vista na Figura 4.12, foi desenhada para ser intuitiva, permitindo que o usuário insira uma pergunta em um campo de texto e visualize a resposta logo após. A página utiliza três colunas principais: uma para exibir o histórico do chat, outra para mostrar as consultas *cypher* geradas, e uma terceira para exibir os resultados retornados pelo banco de dados.

Figura 4.20: Interface do módulo de revisão e crítica dos dados, mostrando as possíveis ações através dos botões de ação. No exemplo, mostra a tabela que possibilita revisar os dados de publicações possivelmente duplicados.

O módulo de revisão e crítica de dados, foi implementado no arquivo `revisoes.py` e pode ser visto na Figura 4.20. Nessa parte do sistema, o usuário pode revisar dados que o sistema identifica como possivelmente incorretos, tais como alunos duplicados ou publicações com títulos semelhantes, utilizando algoritmos de detecção de similaridade.

Foram criados botões específicos que, ao serem acionados, executam consultas no banco de dados *Neo4j* para verificar possíveis inconsistências. Através da interface, o usuário pode visualizar os dados em tabelas e realizar as correções diretamente no banco

de grafos.

Cada botão é responsável por executar uma consulta diferente, como a identificação de alunos com nomes duplicados ou publicações semelhantes, utilizando a função de Jaro Winkler para calcular a similaridade entre os textos. Quando o usuário realiza uma ação de correção, como o *merge* de alunos duplicados, as relações e nós são ajustadas no grafo. Com a construção de uma interface simples, foi possível concentrar o desenvolvimento visando a conclusão dos objetivos do trabalho.

5 Considerações Finais

Este trabalho consistiu na criação de um sistema que modela e analisa dados de atividades docentes utilizando um banco de dados orientado a grafos. A partir da análise de redes complexas, foi possível detectar comunidades acadêmicas e identificar padrões de interação entre docentes. Além disso, o sistema oferece ferramentas para a crítica de dados, permitindo a identificação e correção de inconsistências. A integração de uma interface web interativa e um chat baseado em RAG, suportado por uma LLM, facilita o acesso e a manipulação dos dados pelos usuários, promovendo uma análise fluida das atividades acadêmicas.

5.1 Objetivos específicos atingidos

Primeiramente, foi possível importar e modelar os dados docentes em um banco de grafos, facilitando a detecção de comunidades e a análise de redes complexas. O sistema de crítica dos dados, aliado à interface web interativa, permitiu a correção de inconsistências de forma eficiente, proporcionando uma ferramenta prática para melhorar a qualidade dos dados. Adicionalmente, o chat baseado em RAG oferece uma maneira inovadora de consultar dados e extrair *insights* de forma automatizada, com o auxílio de uma LLM.

Foi atingido o objetivo de importar e modelar os dados docentes em um banco de grafos. A estrutura que foi desenvolvida é flexível e possibilita consultas e análises mais detalhadas tanto sobre as atividades docentes, quanto sobre informações que envolvem alunos, projetos, entre outros aspectos do meio acadêmico. Com a modelagem proposta, também foi possível utilizar técnicas de análise de redes complexas para a detecção de comunidades no grafo.

Também foi possível concluir o desenvolvimento do sistema que utiliza a técnica de RAG, que integra dados do banco de grafos Neo4j com uma LLM através da biblioteca *LangChain*. A implementação incluiu a utilização de um modelo de linguagem da OpenAI, o *gpt-4o* e a aplicação de engenharia de *prompt* para gerar consultas *Cypher* a partir

de perguntas em linguagem natural. O fluxo de funcionamento foi testado e refinado, resultando em um sistema que traduz perguntas do usuário em consultas precisas, executa essas consultas no Neo4j e apresenta respostas compreensíveis em linguagem natural.

Por fim, o desenvolvimento do processo de crítica dos dados foi implementado no projeto, com foco na identificação e correção de inconsistências, como duplicidades de projetos e alunos no banco de dados. Utilizando o algoritmo de *Jaro-Winkler* para detectar variações mínimas nos nomes, o sistema identifica nós duplicados e permite ao usuário revisar e corrigir as entradas por meio de uma interface web. Além disso, foi implementada uma função para verificar nomes de alunos com caracteres especiais ou informações inadequadas, oferecendo uma solução completa para a revisão e correção dos dados. Essas melhorias trouxeram maior integridade no banco de dados e refletiram diretamente nos resultados das análises subsequentes.

5.2 Objetivos específicos não atingidos

Apesar dos avanços, um objetivo específico que não foi atingido foi a implementação de uma raspagem de dados de outras plataformas para enriquecer as fontes de informações. Também não foi possível construir um conjunto de métricas para quantificar a produção, esforço e contribuição, foi possível apenas contabilizar os pontos dos docentes nas atividades. A limitação temporal do projeto foi o principal fator que impediu a realização dessas funcionalidades. No entanto, essa limitação abre portas para melhorias em trabalhos futuros.

5.3 Contribuições do trabalho ao problema específico

O sistema desenvolvido contribui diretamente para a melhoria do processo de avaliação docente ao propor uma solução para a inconsistências nos dados através da detecção e correção de erros, buscando maior precisão na análise das atividades reportadas. Esse processo reduz o esforço necessário para a manutenção de dados consistentes.

Além disso, a combinação da interface web com o chat baseado em RAG oferece uma forma de interação com o sistema que permite que o usuário acesse rapidamente

informações complexas sem a necessidade de conhecimento técnico avançado sobre o banco de dados. Isso aumenta a usabilidade da plataforma e também possibilita a personalização das consultas, de acordo com as necessidades específicas do usuário. Com essa abordagem, o sistema tem o potencial de contribuir com a eficiência do corpo docente, oferecendo uma ferramenta flexível e adaptada às particularidades do ambiente acadêmico.

5.4 Limitações e sugestões para trabalhos futuros

Entre as principais limitações do projeto, destaca-se a impossibilidade de incorporar dados de outras plataformas, o que restringiu a abrangência das análises. Para resolver essa limitação, trabalhos futuros podem focar no desenvolvimento de mecanismos de raspagem de dados em múltiplas fontes, aumentando a riqueza do conjunto de dados e aprimorando a precisão das análises de comunidades e redes complexas. Além disso, melhorias na interface de usuário e no desempenho do sistema podem ser exploradas para tornar o processo de crítica e correção de dados ainda mais fluido e eficiente. A condução de um teste de usabilidade seguido da aplicação do questionário *System Usability Scale* (SUS) pode ser uma proposta interessante para avaliar a usabilidade das soluções propostas, contribuindo para a melhoria contínua da interface e da experiência do usuário.

5.5 Próximos passos

Os próximos passos para dar continuidade ao projeto incluem: a implementação da raspagem de dados em diferentes plataformas, a aplicação de outras estratégias para análise de redes, visando obter percepções mais profundas sobre as relações dos docentes, alunos e dos projetos. A aplicação de um teste de usabilidade, seguida pelo questionário SUS como é mencionado em (KLUG, 2017), também pode ser um passo importante para garantir que as melhorias na interface e na funcionalidade atendam às expectativas dos usuários finais. Além disso, a integração de novas funcionalidades à interface web, como visualizações mais interativas e personalizadas dos dados, pode aprimorar ainda mais a experiência do usuário e a eficácia do sistema como um todo.

Bibliografia

ALIYU, I.; KANA, A. F. D.; ALIYU, S. Development of knowledge graph for university courses management. *International Journal of Education and Management Engineering*, v. 10, n. 2, p. 1–12, 2020. Disponível em: ⟨http://www.mecs-press.org/ijeme/ijeme-v10-n2/IJEME-V10-N2-1.pdf⟩.

AMMAR, A. B. Query optimization techniques in graph databases. *arXiv preprint arXiv:1609.01893*, 2016.

ANGLES, R. The property graph database model. In: *AMW*. [S.l.: s.n.], 2018.

ANGLES, R.; GUTIERREZ, C. Survey of graph database models. *ACM Computing Surveys (CSUR)*, ACM New York, NY, USA, v. 40, n. 1, p. 1–39, 2008.

BLONDEL, V. D.; GUILLAUME, J.-L.; LAMBIOTTE, R.; LEFEBVRE, E. Fast unfolding of communities in large networks. *Journal of statistical mechanics: theory and experiment*, IOP Publishing, v. 2008, n. 10, p. P10008, 2008.

BONIFATI, A.; CIUCANU, R.; LEMAY, A. Learning path queries on graph databases. In: *18th International Conference on Extending Database Technology (EDBT)*. [S.l.: s.n.], 2015.

BRASIL. Lei nº 9.394, de 20 de dezembro de 1996. *Estabelece as diretrizes e bases da*, 1996.

CAPLAN, G. *Support systems and community mental health: Lectures on concept development.* [S.l.]: behavioral publications, 1974.

CARVALHO, D. B. D. *Sistema para Coleta e Avaliação de Relatórios Individuais de Trabalho*. Tese (Doutorado) — Federal University of Juiz de Fora, 2022. Available at http://monografias.ice.ufjf.br/tcc-web/tcc?id=607.

CHANG, Y.; WANG, X.; WANG, J.; WU, Y.; YANG, L.; ZHU, K.; CHEN, H.; YI, X.; WANG, C.; WANG, Y.; YE, W.; ZHANG, Y.; CHANG, Y.; YU, P. S.; YANG, Q.; XIE, X. A survey on evaluation of large language models. *ACM Trans. Intell. Syst. Technol.*, Association for Computing Machinery, New York, NY, USA, v. 15, n. 3, mar 2024. ISSN 2157-6904. Disponível em: ⟨https://doi.org/10.1145/3641289⟩.

COLLARANA, D.; GALKIN, M.; TRAVERSO-RIBÓN, I.; LANGE, C.; VIDAL, M.-E.; AUER, S. Semantic data integration for knowledge graph construction at query time. In: IEEE. *2017 IEEE 11th International Conference on Semantic Computing (ICSC)*. [S.l.], 2017. p. 109–116.

DIRSCHL, C.; KENT, J.; SCHRAM, J.; REUL, Q. Enabling digital business transformation through an enterprise knowledge graph. In: SPRINGER. *The Semantic Web: ESWC 2020 Satellite Events: ESWC 2020 Satellite Events, Heraklion, Crete, Greece, May 31–June 4, 2020, Revised Selected Papers 17*. [S.l.], 2020. p. 298–302.

EXATAS, I. de C. *RESOLUÇÃO N°02/2016*. 2016. Disponível em: ⟨https://www2.ufjf.br/ice/institucional/administrativo-2/resolucoes/⟩.

FERNANDES, D. *Avaliação do desempenho docente: desafios, problemas e oportunidades*. [S.l.]: Texto Editores, 2008.

FEUERRIEGEL, S.; HARTMANN, J.; JANIESCH, C.; ZSCHECH, P. Generative AI. *Business & Information Systems Engineering*, v. 66, n. 1, p. 111–126, fev. 2024. ISSN 1867-0202. Disponível em: ⟨https://doi.org/10.1007/s12599-023-00834-7⟩.

FORTUNATO, S. Community detection in graphs. *Physics reports*, Elsevier, v. 486, n. 3-5, p. 75–174, 2010.

HEIST, N.; HERTLING, S.; RINGLER, D.; PAULHEIM, H. Knowledge graphs on the web-an overview. *Knowledge Graphs for eXplainable Artificial Intelligence*, p. 3–22, 2020.

HOGAN, A.; BLOMQVIST, E.; COCHEZ, M.; D'AMATO, C.; MELO, G. d.; GUTIERREZ, C.; KIRRANE, S.; GAYO, J. E. L.; NAVIGLI, R.; NEUMAIER, S. et al. Knowledge graphs. *ACM Computing Surveys (CSUR)*, ACM New York, NY, USA, v. 54, n. 4, p. 1–37, 2021.

KLUG, B. An overview of the system usability scale in library website and system usability testing. *Weave: Journal of Library User Experience*, Michigan Publishing, University of Michigan Library, v. 1, n. 6, 2017.

KUM, J.; KIM, T.; LEE, M. Evaluating the usability of an llm-aided hybrid avatar agent system. *Journal of Korea Multimedia Society*, 2023.

LAI, Y.-J.; LIU, T.-Y.; HWANG, C.-L. Topsis for modm. *European journal of operational research*, Elsevier, v. 76, n. 3, p. 486–500, 1994.

LATTES. *Sobre a plataforma - Portal Memória*. 2023. Disponível em: ⟨https://memoria.cnpq.br/web/portal-lattes/sobre-a-plataforma⟩.

LEWIS, P. S. H.; PEREZ, E.; PIKTUS, A.; PETRONI, F.; KARPUKHIN, V.; GOYAL, N.; KÜTTLER, H.; LEWIS, M.; YIH, W.; ROCKTÄSCHEL, T.; RIEDEL, S.; KIELA, D. Retrieval-augmented generation for knowledge-intensive NLP tasks. *CoRR*, abs/2005.11401, 2020. Disponível em: ⟨https://arxiv.org/abs/2005.11401⟩.

LI, Z.; WANG, X.; ZHAO, J.; YANG, S.; DU, G.; HU, X.; ZHANG, B.; YE, Y.; LI, Z.; ZHAO, R.; MAO, H. *PET-SQL: A Prompt-Enhanced Two-Round Refinement of Text-to-SQL with Cross-consistency*. 2024. Disponível em: ⟨https://arxiv.org/abs/2403.09732⟩.

MCCARTHY, J. et al. What is artificial intelligence. Stanford University, 2007.

NEO4J. *Cypher Query Language*. 2024. Accessed: 2024-09-13. Disponível em: ⟨https://neo4j.com/docs/cypher-manual/current/introduction/cypher-overview/⟩.

NEWMAN, M. E. Modularity and community structure in networks. *Proceedings of the national academy of sciences*, National Acad Sciences, v. 103, n. 23, p. 8577–8582, 2006.

NEWMAN, M. E. J. The structure and function of complex networks. *SIAM Review*, Society for Industrial Applied Mathematics (SIAM), v. 45, n. 2, p. 167–256, jan. 2003. ISSN 1095-7200. Disponível em: ⟨http://dx.doi.org/10.1137/S003614450342480⟩.

PAULHEIM, H. Knowledge graph refinement: A survey of approaches and evaluation methods. *Semantic web*, IOS Press, v. 8, n. 3, p. 489–508, 2017.

PI, W. The integration of large language models (llms) with neo4j-based knowledge graphs. *arXiv preprint*, 2024. Disponível em: ⟨https://medium.com/@researchgraph/the-integration-of-large-language-models-llms-with-neo4j-based-knowledge-graphs-fe67245bde28⟩.

RAHM, E.; DO, H. H. et al. Data cleaning: Problems and current approaches. *IEEE Data Eng. Bull.*, v. 23, n. 4, p. 3–13, 2000.

REIFSCHNEIDER, M. B. Considerações sobre avaliação de desempenho. *Ensaio: avaliação e políticas públicas em educação*, v. 16, n. 58, p. 47–58, 2008.

REYNOLDS, L.; MCDONELL, K. Prompt programming for large language models: Beyond the few-shot paradigm. In: *Extended Abstracts of the 2021 CHI Conference on Human Factors in Computing Systems*. New York, NY, USA: Association for Computing Machinery, 2021. (CHI EA '21). ISBN 9781450380959. Disponível em: ⟨https://doi.org/10.1145/3411763.3451760⟩.

ROBINSON, I.; WEBBER, J.; EIFREM, E. *Graph Databases*. [S.l.]: O'Reilly Media, Inc., 2013.

TEAM, C. *What is Cytoscape?* 2024. Accessed: 2024-09-13. Disponível em: ⟨https://cytoscape.org/what_is_cytoscape.html⟩.

TEIXEIRA, E. B. A análise de dados na pesquisa científica: importância e desafios em estudos organizacionais. *Desenvolvimento em questão*, v. 1, n. 2, p. 177–201, 2003.

TUIJN, C.; GYSSENS, M. Cgood, a categorical graph-oriented object data model. *Theoretical Computer Science*, Elsevier, v. 160, n. 1-2, p. 217–239, 1996.

UFJF. *Acesso ao SIGA*. 2023. Disponível em: ⟨https://www.ufjf.br/bach/utilidades/acesso-ao-siga/⟩.

WANG, C.; GAN, T.; LI, X.; ZHANG, L.; WANG, X. An ontology-enhanced knowledge graph embedding method. In: *Proceedings of the 2023 12th International Conference on Computing and Pattern Recognition*. New York, NY, USA: Association for Computing Machinery, 2024. (ICCPR '23), p. 51–57. ISBN 9798400707988. Disponível em: ⟨https://doi.org/10.1145/3633637.3633645⟩.

WANG, Y.; QIN, J.; WANG, W. Efficient approximate entity matching using jaro-winkler distance. In: BOUGUETTAYA, A.; GAO, Y.; KLIMENKO, A.; CHEN, L.; ZHANG, X.; DZERZHINSKIY, F.; JIA, W.; KLIMENKO, S. V.; LI, Q. (Ed.). *Web Information Systems Engineering – WISE 2017*. Cham: Springer International Publishing, 2017. p. 231–239. ISBN 978-3-319-68783-4.

WANKE, P. F.; ANTUNES, J. J.; MIANO, V. Y.; COUTO, C. L. do; MIXON, F. G. Measuring higher education performance in brazil: government indicators of performance vs ideal solution efficiency measures. *International Journal of Productivity and Performance Management*, Emerald Publishing Limited, v. 71, n. 6, p. 2479–2495, 2021.

WHITEBOOK, M. Early education quality: Higher teacher qualifications for better living environments. a review of the literature. ERIC, 2003.

WOHLIN, C. Guidelines for snowballing in systematic literature studies and a replication in software engineering. In: *Proceedings of the 18th international conference on evaluation and assessment in software engineering*. [S.l.: s.n.], 2014. p. 1–10.

ZHOU, X.; ZHAO, X.; LI, G. *LLM-Enhanced Data Management*. 2024. Disponível em: ⟨https://arxiv.org/abs/2402.02643⟩.