UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
CURSO DE ENGENHARIA DE COMPUTAÇÃO

EDUARDO SPITZER FISCHER

VizColab: Visualização de uma rede de colaboração acadêmica brasileira de larga escala gerada a partir de dados da CAPES

Monografia apresentada como requisito parcial para a obtenção do grau de Bacharel em Engenharia da Computação

Orientador: Prof. Dr. Juliano Araújo Wickboldt

Porto Alegre
2022

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
Reitor: Prof. Carlos André Bulhões
Vice-Reitora: Profª. Patricia Pranke
Pró-Reitora de Graduação: Profª. Cíntia Inês Boll
Diretora do Instituto de Informática: Profª. Carla Maria Dal Sasso Freitas
Coordenador do Curso de Engenharia de Computação: Prof. Walter Fetter Lages
Bibliotecário-chefe do Instituto de Informática: Alexsander Borges Ribeiro

“The truth is not for all men
but only for those who seek it.”
— Ayn Rand

AGRADECIMENTOS

Aos meus pais, Claudiomir Fischer e Rosângela Spitzer Fischer, e à minha irmã, Nathália Spitzer Fischer, que nunca mediram esforços para me apoiar ao decorrer de toda a minha jornada. Eles são a base sólida que me permitiu chegar até aqui e que me inspira e me dá confiança para almejar conquistas cada vez maiores. Muito obrigado, serei eternamente grato.

Aos meus colegas de curso, em especial Lando, Rodrigo e Probst (que eventualmente nos trocou pela matemática), que desde o início dessa jornada se tornaram excelentes amigos que pretendo manter para o resto da vida. Vocês compreendem como ninguém o tamanho do desafio que assumimos ao iniciar o curso de Engenharia de Computação nesta universidade. Olhando para trás, percebo que a nossa união ao enfrentarmos juntos cada uma das dificuldades foi componente fundamental da força que me fez persistir nessa jornada.

A todos os meus amigos com quem compartilhei alegrias e angústias, vivi momentos memoráveis e de onde obtive suporte sempre que precisei, muito obrigado. Agradeço especialmente ao Guilherme e ao Lucca, que sempre estiveram ao meu lado e me acompanharam desde a Escola de Ensino Fundamental em São Lourenço do Sul até a universidade em Porto Alegre.

A cada um dos professores com quem tive o prazer de aprender em cada etapa desse curso. Obrigado pela dedicação, paciência e pela generosidade com as quais transmitem o seu conhecimento. Se fazemos parte de uma universidade de excelência, isso se deve à paixão e ao comprometimento com os quais vocês transitam as fronteiras do conhecimento.

Por fim, agradeço ao meu orientador, Prof. Dr. Juliano Wickboldt, pelo comprometimento em me acompanhar durante este último ano ao longo do desenvolvimento deste projeto. Cada uma das suas ideias, orientações e críticas foi fundamental no caminho que percorremos entre a nossa ideia inicial até a conclusão deste trabalho.

RESUMO

Redes de colaboração acadêmica consistem em grafos onde pesquisadores são modelados como nós enquanto co-autorias são modeladas como arestas entre pesquisadores. Enquanto a análise e a visualização dessas redes pode ser feita facilmente para pequenos grafos, fazê-las para redes de larga escala expõe diversos desafios, especialmente na matéria de visualização destes grafos. Este trabalho realiza, em um primeiro momento, uma revisão da literatura relacionada a análise e visualização de redes de co-autoria acadêmica e explora ferramentas existentes para visualização, manipulação e armazenamento de grafos. Isso é feito com a finalidade de traçar um panorama do estado da arte deste domínio e verificar a viabilidade de uso das técnicas e ferramentas estudadas na composição de uma solução para o problema da geração e visualização de uma rede de colaborações acadêmicas brasileira de larga escala. Em seguida, uma rede de colaborações acadêmicas em programas de pós-graduação *stricto sensu* brasileiros é gerada a partir de conjuntos de dados mantidos pela CAPES e publicados quadrienalmente no portal Dados Abertos CAPES (CAPES, 2022a), sobre os quais são aplicadas técnicas de sanitização de dados, agrupamento de entidades semelhantes e enriquecimento. O resultado é um grafo composto de 532 universidades brasileiras, 4.685 programas de pós-graduação *stricto sensu*, 1.275.852 autores, 1.708.666 produções acadêmicas e 14.883.507 relações de co-autoria de trabalhos acadêmicos. Os dados processados são então modelados e importados em uma base de dados de grafos, permitindo uma ampla gama de consultas personalizadas de forma eficiente e sob demanda. Com a finalidade de possibilitar a exploração dinâmica dessa rede, é desenvolvido o software VizColab, uma aplicação web que permite a visualização dos dados de colaborações acadêmicas na forma de grafos tri-dimensionais em três níveis hierárquicos distintos: universidades, programas de pós-graduação e autores de produções acadêmicas. A aplicação implementa técnicas como a segmentação hierárquica de nós e introduz o conceito de densidade de conexões, o que permite a ocultação de arestas menos significativas, proporcionando ao usuário uma experiência de visualização interativa, clara e concisa da rede de colaborações.

Palavras-chave: Colaborações acadêmicas. co-autorias acadêmicas. visualização. grafos. análise de grafos. redes de co-autoria. pós graduação. CAPES.

VizColab: Visualizing a large-scale Brazillian academic collaboration network generated from CAPES data

ABSTRACT

Academic collaboration networks are graphs where researchers are modeled as nodes while co-authorships are modeled as links between researchers. Although the analysis and visualization of such networks can be easily implemented for small graphs, doing so for large-scale graphs introduces a handful of challenges, especially regarding the subject of visualizing such graphs. This study conducts, at first, a review of the literature related to academic co-authorship networks analysis and visualization and explores existing tools for graph visualization, manipulation, and storage, intending to get an overview of the domain's state-of-the-art, as well as verify the viability of using the considered tools and techniques in the design of a solution to the problem of generating and visualizing a Brazillian large-scale academic collaboration network. Subsequently, an academic collaboration network is generated based on Brazillian *stricto sensu* graduate programs data mantained by CAPES and published every four years in the Dados Abertos CAPES portal (CAPES, 2022a), over which data sanitization, similar entities grouping, and data enrichment techniques are applied. The process results in a graph containing 532 universities, 4,685 *stricto sensu* graduate programs, 1,275,852 authors, 1,708,666 intellectual productions and 14,883,507 co-authorship links between authors. The processed data is then modeled and imported into a graph database, allowing a wide range of custom queries to be executed efficiently and on-demand. To allow dynamic exploration of the network, the VizColab software is created. A web application that allows academic collaboration data to be visualized in the form of three-dimensional graphs in three distinct hierarchic levels: universities, graduate programs, and intellectual production authors. The application implements techniques such as hierarchic segmentation and introduces the concept of least-significant links screening, offering the user an interactive, clear, and seamless visualization experience.

Keywords: academic collaborations, academic co-authorships, visualization, graphs, graph analysis, co-authorship networks, graduate, CAPES.

LISTA DE ABREVIATURAS E SIGLAS

ABNT	Associação Brasileira de Normas Técnicas
CAPES	Coordenação de Aperfeiçoamento de Pessoal de Nível Superior
DBLP	Digital Bibliography and Library Project
e-MEC	Sistema eletrônico de acompanhamento de processos do Ministério da Educação
IES	Instituição de Ensino Superior
PPGC	Programa de Pós-Graduação em Computação
RAM	Random-Access Memory (Memória de acesso aleatório)

LISTA DE FIGURAS

Figura 2.1 Processo empregado em Silva Junior et al. (2022) para obtenção do grafo de colaborações 16
Figura 2.2 Grupos de pesquisa identificados em Perianes-Rodríguez, Olmeda-Gómez and Moya-Anegón (2010) 16
Figura 2.3 Interface *PostHistory* (VIéGAS; DONATH, 2004) com um painel de calendário à esquerda e painel de contatos à direita. 18
Figura 2.4 Modelo de visualização de grafos *ClusterVis* (CAVA; FREITAS; WINCKLER, 2017). O exemplo corresponde a uma rede de co-autorias acadêmicas, onde a cor dos círculos representa um atributo categórico (professor, aluno ou externo), as barras representam atributos numéricos (*papers* em *journals*, capítulos em livros e *papers* em conferências) e as arestas ao centro representam relações de co-autoria entre os autores. 19
Figura 2.5 Grafo de um subconjunto de dados a respeito de vendas de produtos no website amazon.com (DEVI; KASIREDDY, 2019). Os nós representam produtos, e arestas entre dois nós representam produtos que costumam ser comprados em conjunto 20
Figura 2.6 Imãs virtuais da ferramenta *MagnetViz* (SPRITZER; FREITAS, 2011) 20
Figura 2.7 Esquemático da visualização adaptativa proposta em Miyamura et al. (2011). 21
Figura 2.8 Visualização de um grafo de co-autorias do PPGC UFRGS (WICKBOLDT, 2019) utilizando a biblioteca 3D Force Graph (ASTURIANO, 2017).... 21
Figura 2.9 Consulta efetuada na base de dados Neo4j requisitando todos os nós que possuem o relacionamento CO_AUTOR com o Autor de nome JULIANO ARAUJO WICKBOLDT dentro de uma rede composta por autores e produções acadêmicas dos anos de 2017 a 2019 (CAPES, 2022a). A consulta em questão levou 5ms para ser concluída. 22
Figura 2.10 Variação do tempo de importação e da taxa de produções importadas por segundo para *datasets* de diferentes tamanhos. 23

Figura 3.1 Esquemático ilustrando a arquitetura da solução implementada 26

Figura 4.1 Aplicação VizColab - Visualização dos autores do programa de computação da UFRGS 52
Figura 4.2 Arquitetura de alto nível da aplicação VizColab 53
Figura 4.3 Ilustração do crescimento exponencial do número de conexões em decorrência do aumento linear do número de nós em um grafo completo 54
Figura 4.4 Aplicação VizColab — Visualização ao nível de colaborações entre instituições de ensino superior brasileiras 56
Figura 4.5 Aplicação VizColab — Visualização ao nível de colaborações entre programas de pós graduação da UFRGS 57
Figura 4.6 Aplicação VizColab — Visualização ao nível de colaborações entre autores ligados ao programa de computação da UFRGS 58
Figura 4.7 Aplicação VizColab — Visualização de co-autorias entre autores que colaboram com o docente Juliano Araújo Wickboldt 59
Figura 4.8 À direita, o grafo de colaborações entre instituições de ensino contendo todas as arestas; À esquerda, o mesmo grafo com o uso de um parâmetro de densidade de conexões de valor 3. 60

Figura 4.9 Recursos de exploração da ferramenta VizColab 62

Figura 5.1 Visualização de instituições de ensino superior no software VizColab, destacando os núcleos regionais. 65
Figura 5.2 Acima, visualização dos programas de pós-graduação da USP, uma instituição generalista; Abaixo, visualização dos programas de pós-graduação da UFCSPA e ITA, instituições especialistas. 67
Figura 5.3 Visualização de autores centrais identificados no programa de computação da UFRGS. 68

LISTA DE TABELAS

Tabela 2.1 Tempo de importação e taxa de importação de dados de autoria de produções intelectuais para o banco de dados Neo4J para diferentes volumes de dados 23

Tabela 3.1 Resultados de cada uma das etapas de agrupamento de autores 37

SUMÁRIO

1 INTRODUÇÃO 12
2 TRABALHOS RELACIONADOS E FERRAMENTAS EXISTENTES 15
2.1 Análise de redes sociais de colaboração acadêmica 15
2.2 Visualização de grafos 17
2.3 Ferramentas e *frameworks* 20
2.3.1 3D Force Graph (ASTURIANO, 2017) 20
2.3.2 Neo4J (NEO4J, 2022a) 22
3 GERAÇÃO DA REDE DE COLABORAÇÃO ACADÊMICA 26
3.1 Obtenção de Dados 26
3.2 Processamento de Dados 30
3.2.1 Conjunto de Dados de Autor da Produção Intelectual 30
3.2.2 Conjunto de Dados de Produções Intelectuais 39
3.2.3 Conjunto de Dados de Programas da Pós-Graduação 41
3.2.4 Conjunto de Dados de Cursos da Pós-Graduação 42
3.2.5 Pós-Processamento de dados 44
3.3 Banco de Dados de Grafos 45
3.3.0.1 Nós 45
3.3.0.2 Relacionamentos 47
4 APLICAÇÃO DE VISUALIZAÇÃO DA REDE 52
4.1 Arquitetura da aplicação 53
4.2 Técnicas para a visualização do grafo de larga escala 53
4.2.1 Segmentação hierárquica de nós 54
4.2.2 Densidade de conexões variável 57
4.3 Elementos visuais do grafo 59
4.4 Exploração da rede de colaborações 62
5 ANÁLISE DA SOLUÇÃO 64
5.1 Identificação de núcleos regionais de pesquisa 64
5.2 Relevância nacional ou local de instituições de ensino 65
5.3 Focos de pesquisa de instituições de ensino 66
5.4 Identificação de autores centrais dentro de programas de pós graduação 67
6 CONSIDERAÇÕES FINAIS E TRABALHOS FUTUROS 70
REFERÊNCIAS 71

1 INTRODUÇÃO

O conceito de redes de colaboração acadêmica vem sendo discutido há décadas, baseado na ideia de que pesquisadores podem ser modelados como nós de um grafo, enquanto co-autorias – condição em que dois pesquisadores são listados como autores em um mesmo trabalho – podem ser modeladas como arestas conectando os dois nós. A rede resultante representa indivíduos que colaboraram entre si em prol do desenvolvimento de conhecimento e carrega potencial para a extração de informações valiosas a respeito da influência da colaboração no resultado das produções acadêmicas. O uso de co-autorias em produções acadêmicas como forma de medição da colaboração entre pesquisadores tem sido um tópico de interesse desde os anos 1960 (KUMAR, 2015), de forma que co-autorias podem hoje ser consideradas um *proxy* confiável para colaborações acadêmicas.

As principais pesquisas nesse campo de estudos se dividem em duas vertentes: análise de redes de colaboração e visualização de redes de colaboração. A análise de redes de co-autorias propõe utilizar métodos matemáticos e de teoria dos grafos para sintetizar estatísticas e correlações relevantes a respeito de pesquisadores, grupos de pesquisa, universidades, programas de pós graduação, entre outros, contando principalmente com o uso de métricas bem conhecidas do campo de estudos denominado *Social Network Analysis* (KUMAR, 2015). Já a visualização de redes de colaboração acadêmica diz respeito ao uso de técnicas de visualização de grafos para exibir o conjunto de dados de forma coesa e amigável a usuários, possibilitando a análise visual e extração de *insights* decorrentes da inspeção visual dos nós e seus relacionamentos na rede.

Tanto a análise quanto a visualização dessas redes podem ser feitas facilmente para pequenos grupos de pesquisadores. Entretanto, quando aumentamos a escala do volume de dados analisados para agregar pesquisadores e produções de diferentes programas de pós-graduação, de uma universidade inteira, ou até mesmo de um país inteiro, barreiras técnicas são introduzidas. Características intrínsecas deste tipo de rede fazem com que o número de conexões entre pesquisadores aumente de forma exponencial comparado ao aumento do número de pesquisadores inseridos na base de dados. Isso se torna um problema especialmente na matéria da visualização destas redes, já que o conjunto de dados juntamente com seus elementos gráficos rapidamente se tornam grandes demais para serem mantidos e manipulados em memória, ao mesmo tempo que uma grande quantidade de nós e arestas sobrepostas dificulta a visualização do conteúdo de interesse. Uma instância deste problema de redes de colaboração de larga escala pode ser observada na

rede de co-autorias acadêmicas extraída a partir dos dados disponíveis na plataforma Dados Abertos CAPES (CAPES, 2022a), que conta com dados relacionados a produções intelectuais autoradas por pesquisadores de todos os programas de pós-graduação *stricto sensu* do Brasil.

Em um primeiro momento, este trabalho realiza um estudo da literatura existente nas áreas de análise de redes de co-autoria e visualização de grafos, com a finalidade de identificar propostas existentes para solucionar problemas intrínsecos à análise e visualização de grafos de larga escala, bem como avaliar ferramentas já existentes no mercado com potencial para compor uma possível solução para o problema. O trabalho também analisa diferentes conjuntos de dados disponíveis na plataforma de Dados Abertos da CAPES (CAPES, 2022a), identificando aqueles com dados relevantes para a construção de uma rede de co-autorias. Em um segundo momento, uma rede de colaboração acadêmica de âmbito nacional é concebida a partir de dados disponíveis na plataforma Dados Abertos CAPES (CAPES, 2022a), incluindo mais de 1,2 milhões de autores distribuídos entre 4.685 programas de pós-graduação *stricto sensu* brasileiros. Em seguida, é desenvolvida uma solução denominada VizColab – uma aplicação web interativa que possibilita a visualização dinâmica desta rede através de representações gráficas tri-dimensionais – permitindo que usuários explorem dados de colaborações acadêmicas em três níveis distintos: universidades, programas de pós-graduação e autores.

As seções restantes deste documento são organizadas da seguinte forma: na seção 2 é feita uma revisão da literatura nas áreas de análise de redes de colaboração acadêmica e visualização de grafos com a finalidade de identificar os principais trabalhos existentes neste domínio. Esta seção também experimenta ferramentas e *frameworks* existentes relacionados a visualização, processamento e armazenamento de grafos com o objetivo de traçar um panorama do estado da arte deste domínio e verificar a viabilidade do uso destas ferramentas na composição da solução proposta. A seção 3 apresenta a primeira parte da solução desenvolvida neste trabalho, discorrendo sobre os recursos e metodologias empregados para a concepção de uma rede de colaborações acadêmicas de escala nacional gerada a partir de *datasets* disponíveis no portal de Dados Abertos da CAPES (CAPES, 2022a). A seção 4 apresenta a segunda parte da solução desenvolvida neste trabalho, detalhando a construção do VizColab, uma aplicação desenvolvida para a visualização tri-dimensional e exploração dinâmica dos dados da rede de co-autorias de larga escala concebida na etapa anterior. A seção 5 faz uma análise da ferramenta VizColab, produto final deste trabalho, demonstrando e avaliando sua aplicação em potenciais casos de uso

de forma a demonstrar como a ferramenta pode agregar valor aos seus usuários. A seção 6, por fim, apresenta as considerações finais deste trabalho e discorre sobre possíveis melhorias contempladas para trabalhos futuros.

2 TRABALHOS RELACIONADOS E FERRAMENTAS EXISTENTES

Essa seção analisa os principais trabalhos relacionados à análise de redes de colaboração acadêmica e visualização de grafos, relatando o estado da arte desses campos de pesquisa.

2.1 Análise de redes sociais de colaboração acadêmica

Redes sociais são um campo de estudos que tem atraído a atenção de pesquisadores há décadas. Alguns estudos deste tópico datam do início dos anos 1930, quando eram realizados predominantemente por antropólogos e sociólogos (SILVA JUNIOR et al., 2022). Atualmente, a combinação entre o estudo de redes sociais e o campo de teoria dos grafos tem gerado resultados interessantes. Um dos resultados dessa intersecção é o foco de estudo que propõe identificar correlações entre redes sociais de co-autorias em trabalhos acadêmicos e os resultados qualitativos obtidos por esses pesquisadores em suas publicações.

Sendo um dos trabalhos que explora essa temática, Silva Junior et al. (2022) propõem identificar correlações entre a avaliação CAPES de programas de pós-graduação brasileiros e padrões topológicos identificados em redes de colaboração acadêmica. Para isso, os autores construíram um grafo de co-autorias em trabalhos acadêmicos baseado em dados de 1644 pesquisadores da área de ciência da computação obtidos através das plataformas Sucupira e Lattes. Neste grafo, autores de trabalhos são modelados como nós, enquanto co-autorias (pesquisadores listados como autores em um mesmo trabalho) são modeladas como arestas. Em seguida, foram calculadas um total de 42 métricas complexas a respeito do grafo como número de nós, número de arestas, *betweenness centrality*, *cluster coefficient*, dentre outras. Essas métricas serviram para compor uma matriz de *features*, sobre a qual algoritmos para identificação de correlações foram executados a fim de obter aqueles com maior correlação com o desempenho CAPES de um programa de pós graduação. A figura 2.1 apresenta um diagrama descrevendo o processo empregado pelos autores para a geração do grafo.

Já Perianes-Rodríguez, Olmeda-Gómez and Moya-Anegón (2010) propõem uma técnica baseada em análise fatorial para identificar grupos de pesquisa em rede de colaboração acadêmica. Os autores analisam um *dataset* com trabalhos de pesquisadores de 9 departamentos da Universidade Carlos III de Madrid e defendem a ideia de que sub-

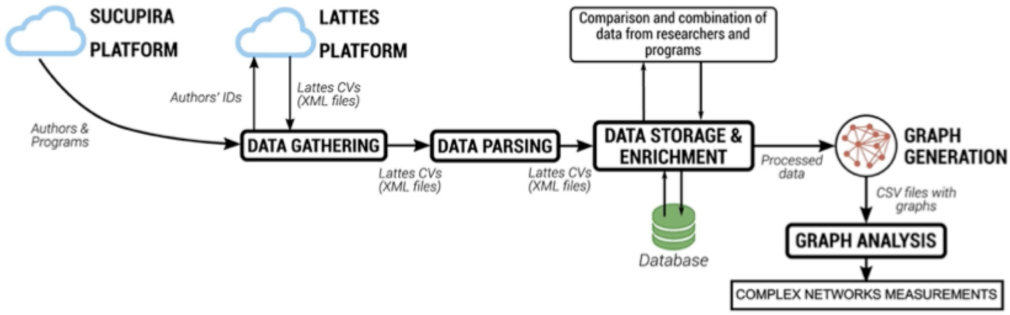


Figura 2.1 – Processo empregado em Silva Junior et al. (2022) para obtenção do grafo de colaborações

conjuntos de nós fortemente relacionados no grafo de co-autorias têm alta probabilidade de representar grupos de pesquisa e utilizam a técnica de análise fatorial juntamente com dados de categorias do *Journal Citation Reports* para identificar e classificar os grupos de pesquisa encontrados.

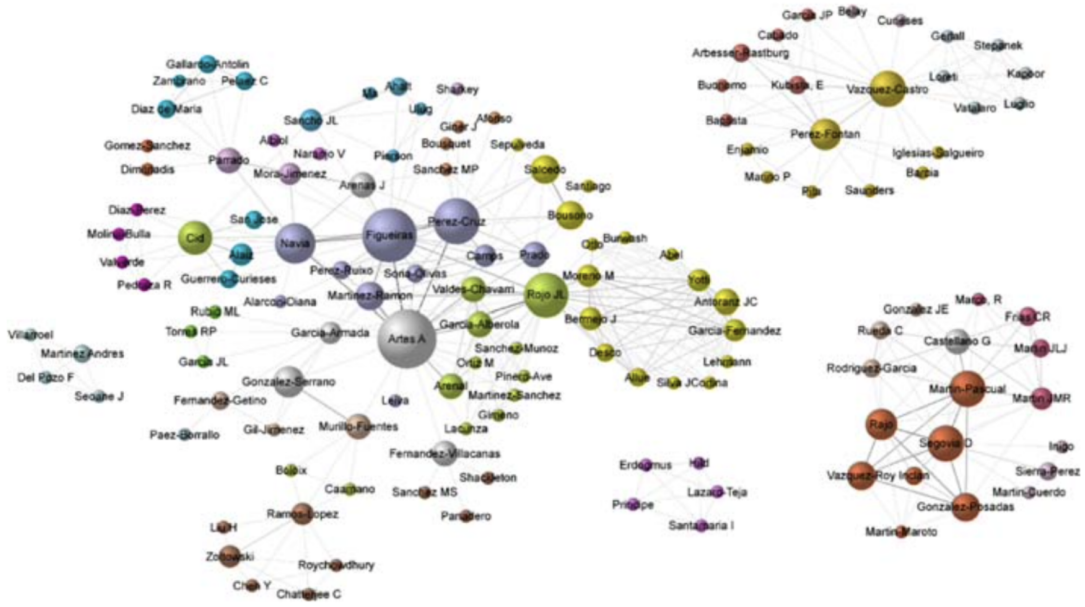


Figura 2.2 – Grupos de pesquisa identificados em Perianes-Rodríguez, Olmeda-Gómez and Moya-Anegón (2010)

Ainda no contexto de identificação de grupos, Aggrawal and Arora (2016) utilizou o software Gephi para analisar o *dataset Digital Bibliography and Library Project* (DBLP), um grafo de 17.280 nós e 58.539 arestas, sendo os nós pesquisadores científicos e as arestas co-autorias entre 2 pesquisadores em um trabalho acadêmico. Os autores utilizaram o layout Force Atlas 2 do software Gephi para detectar comunidades, que são definidas como grupos de autores que colaboram entre si com frequência. Além da detecção de comunidades, este trabalho ainda analisa estatísticas a respeito da rede de colaboração sob duas ópticas distintas: análise global, que avalia as métricas de distribuição de graus dos nós e seu coeficiente de agrupamento; e análise local, que avalia as métricas de

betweeness centrality e *closeness centrality*.

Já o trabalho Kumar (2015) faz uma revisão da literatura acadêmica relacionada a redes de co-autoria em trabalhos acadêmicos. Neste artigo, o autor explora e discute trabalhos existentes a respeito de conceitos como *Small World*, *Giant Components*, aspectos de redes de co-autoria, ambiguidade de nomes de autores, visualização de redes, análise de redes multi-nível, comunidades acadêmicas, assortatividade, estudos comparativos, comportamento em citações de co-autorias e correlações entre medidas de centralidade e produtividade. Este artigo apresenta uma ampla abrangência dentro dos assuntos de interesse da presente pesquisa, tendo servido como base teórica para muitos dos desafios encontrados durante o desenvolvimento deste trabalho.

2.2 Visualização de grafos

O outro campo de pesquisa de interesse desse trabalho diz respeito à visualização de redes. Aqui foram estudados trabalhos que versam sobre visualizações interativas, layouts de visualização alternativos, visualização de redes de larga escala, comunicatividade de grafos, dentre outros temas.

Dentro da temática de visualização interativa, Viégas and Donath (2004) sugerem que visualizações destinadas a usuários devem ir além do paradigma de grafos, incorporando princípios cartográficos básicos como zoom adaptativo e múltiplos modos de visualização. Além disso, é proposta também uma visualização alternativa nomeada *PostHistory*, que demonstra também a dimensão temporal dos dados que compõem a rede de nós e foi desenvolvida para ser utilizada em conjunto com uma visualização de grafos tradicional (Figura 2.3). Para demonstrar essa proposta, os autores criaram uma visualização para um conjunto de dados baseados em trocas de mensagens de e-mail.

Ainda sobre visualizações alternativas, Cava, Freitas and Winckler (2017) introduzem o modelo de *ClusterVis* (Figura 2.4), desenhado para a visualização de nós que compõem um *cluster* de um grafo. Essa visualização permite que diversos atributos de cada um dos nós sejam visualizados simultaneamente, sem *overlaps*, enquanto preserva informações de relacionamento entre os nós (arestas).

Já no contexto da análise de grandes quantidades de dados, Devi and Kasireddy (2019) sugerem que visualizações gráficas, juntamente com métricas a respeito desses grafos, são uma forma de dar sentido a grandes quantidades de dados coletados a todo momento por empresas (*big data*). Os autores focam então em criar, transformar, visu-

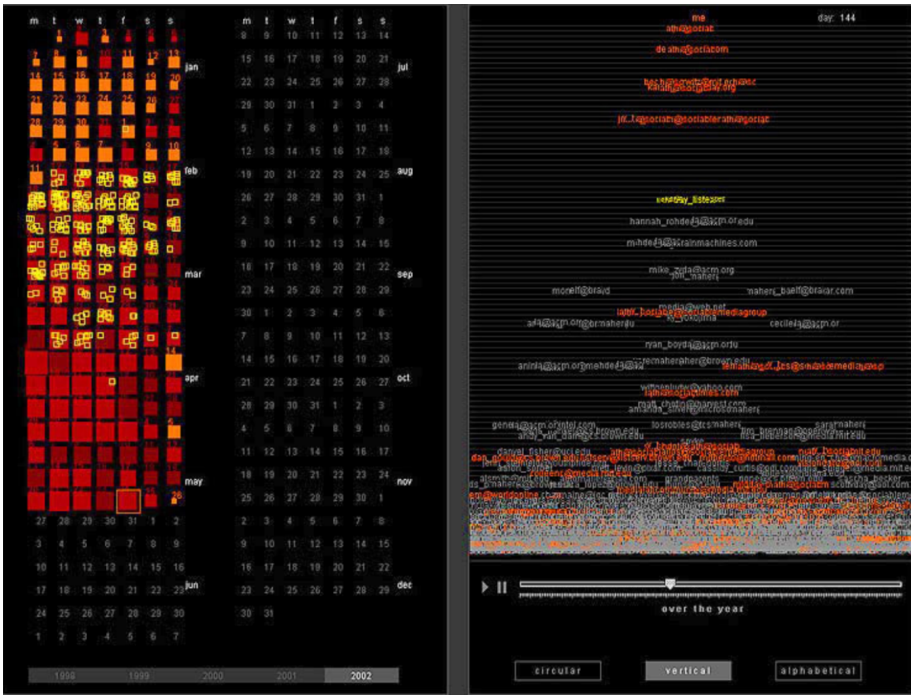


Figura 2.3 – Interface *PostHistory* (VIéGAS; DONATH, 2004) com um painel de calendário à esquerda e painel de contatos à direita.

alizar e analisar um grafo de larga escala baseado em dados de compras de produtos no site amazon.com (Figura 2.5). A análise do grafo resultante ajudou a responder perguntas a respeito de lucratividade de produtos, efetividade de sugestões em carrinhos de compras, entre outros, provando ser uma ferramenta útil para agregar valor de negócio a um conjunto de dados.

O trabalho Spritzer and Freitas (2011) propõe a ferramenta *MagnetViz* (Figura 2.6, que introduz às representações de grafos o uso de imãs virtuais capazes de exercer forças de atração ou repulsão sobre determinados nós de um grafo. Sempre que um novo imã é inserido, o software é capaz de reorganizar as forçar em ação para refletir as mudanças.

Em Spritzer et al. (2015), é feita uma revisão da literatura, analisando trabalhos relacionados a grafos comunicativos, editores de grafos existentes e manipulação de layouts interativos. Em seguida, identifica os elementos básicos de um diagrama nós-arestas comunicativo estático, derivando seis tarefas que um usuário precisa desempenhar para criar um desses diagramas. A partir disso, os autores introduzem o protótipo *GraphCoiffure*, que propõe ser uma ponte entre editores gráficos e softwares de criação de grafos, possuindo o intuito de melhorar o poder comunicativo e deixar os diagramas mais conformantes ao aspecto estético desejado.

Em Miyamura et al. (2011), os autores propõem uma técnica de visualização adaptativa para *datasets* hierárquicos. A técnica seleciona um estilo de grafo adequado de

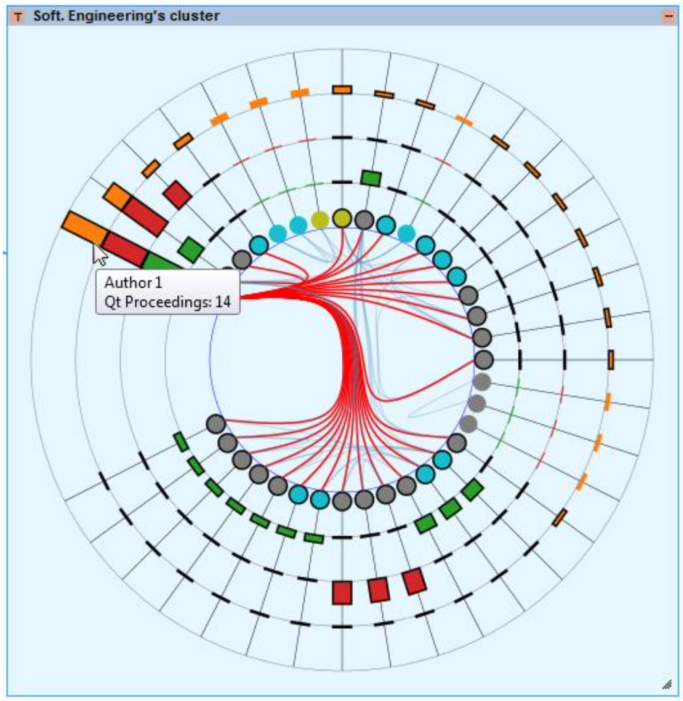


Figura 2.4 – Modelo de visualização de grafos *ClusterVis* (CAVA; FREITAS; WINCKLER, 2017). O exemplo corresponde a uma rede de co-autorias acadêmicas, onde a cor dos círculos representa um atributo categórico (professor, aluno ou externo), as barras representam atributos numéricos (*papers* em *journals*, capítulos em livros e *papers* em conferências) e as arestas ao centro representam relações de co-autoria entre os autores.

acordo com a densidade de nós em uma determinada área. São apresentados dois estilos: *Node-Link* e *Space-Filling*. Para áreas do grafo onde há espaço livre, é utilizada a visualização Node-Link, onde nós e arestas são desenhados normalmente. Para áreas densas, é adotada a visualização *Space-Filling*, onde arestas são ocultadas e nós são representados por *pixels* únicos. Já para regiões muito densas, onde o tamanho de um nó é inferior ao tamanho de um pixel, é adotado o modelo *Space-Filling* com uma simplificação dos nós. O resultado pode ser visualizado na Figura 2.7.

Por fim, ainda no tema de visualizações adaptativas, o trabalho de Shi et al. (2009) tem a finalidade de criar visualizações de redes sociais compreensíveis em qualquer escala, topologia e tamanho de tela, com ferramentas de navegação que possibilitem a análise de todos os detalhes da rede, animações suaves e com tempo de resposta rápido. Para atingir esse objetivo, os autores propõem o uso de técnicas como ocultação de nós menos importantes, carregamento adaptativo de dados, *caching*, animações para mascarar o carregamento de dados, pré-processamento de nós e algoritmo baseado em forças para disposição dos nós no grafo.

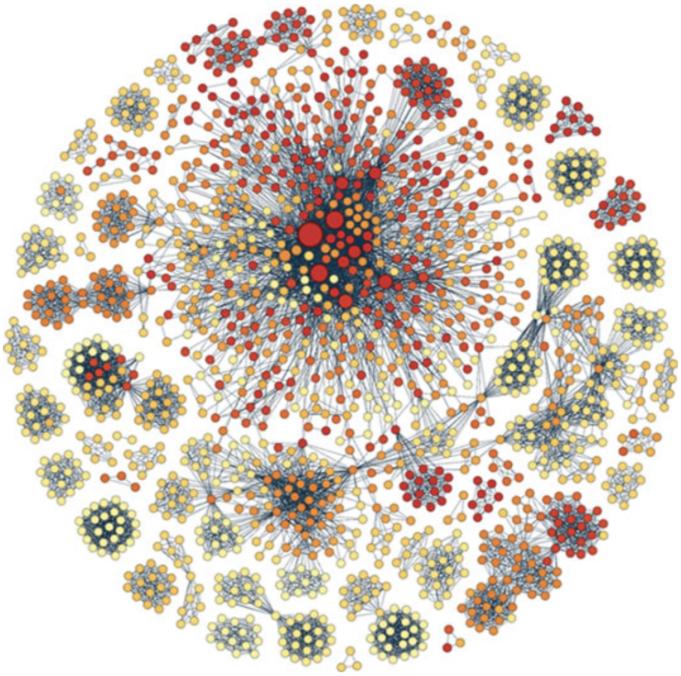
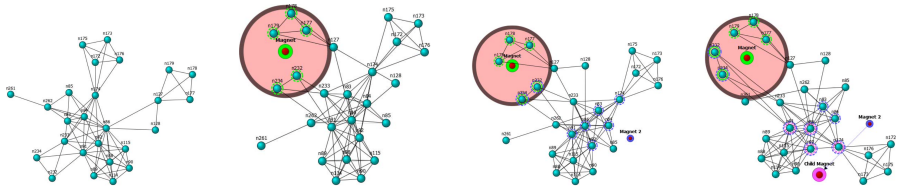


Figura 2.5 – Grafo de um subconjunto de dados a respeito de vendas de produtos no website amazon.com (DEVI; KASIREDDY, 2019). Os nós representam produtos, e arestas entre dois nós representam produtos que costumam ser comprados em conjunto



(a) Original layout. (b) A simple attraction. (c) Magnet intersection. (d) Magnet hierarchy.

Figura 2.6 – Imãs virtuais da ferramenta *MagnetViz* (SPRITZER; FREITAS, 2011)

2.3 Ferramentas e *frameworks*

Esta seção analisa ferramentas e *frameworks* de visualização, análise, armazenamento e consulta de grafos disponíveis para uso. As principais ferramentas foram selecionadas para a realização de testes com a finalidade de avaliar sua aplicabilidade no presente trabalho.

2.3.1 3D Force Graph (ASTURIANO, 2017)

Esta biblioteca *JavaScript* permite a visualização de grafos utilizando uma abordagem interativa e tri-dimensional. Para a geração do layout do grafo, é utilizado um modelo de forças aplicado sobre os nós da rede, onde forças de atração são aplicadas entre dois nós conectados por uma aresta (de forma análoga ao efeito mola) enquanto forças

(a) Node-Link tree graph. (b) Space-Filling graph.
Fig. 4: Selective tree graph styles.

Fig. 5: Simplification of the tree graph in the area in which the node density is high.

Figura 2.7 – Esquemático da visualização adaptativa proposta em Miyamura et al. (2011).

de repulsão são aplicadas a todos os nós (de forma análoga a força nuclear nos átomos). O usuário pode interagir com o grafo por meio de movimentos de câmera como rotação, translação e zoom. A ferramenta também trata da re-organização da rede em tempo real em resposta a deslocamentos nos nós.

Para entender melhor o potencial da biblioteca, foi criada uma visualização a partir de um conjunto de dados de co-autorias do Programa de Pós-Graduação em Computação da UFRGS (PPGC-UFRGS) (WICKBOLDT, 2019) (Figura 2.8). A visualização foi testada com diferentes volumes de dados, sendo possível observar que a biblioteca é capaz de oferecer uma experiência de visualização fluida e responsiva para a exibição de um conjunto de dados de até 1000 nós e 4000 arestas.

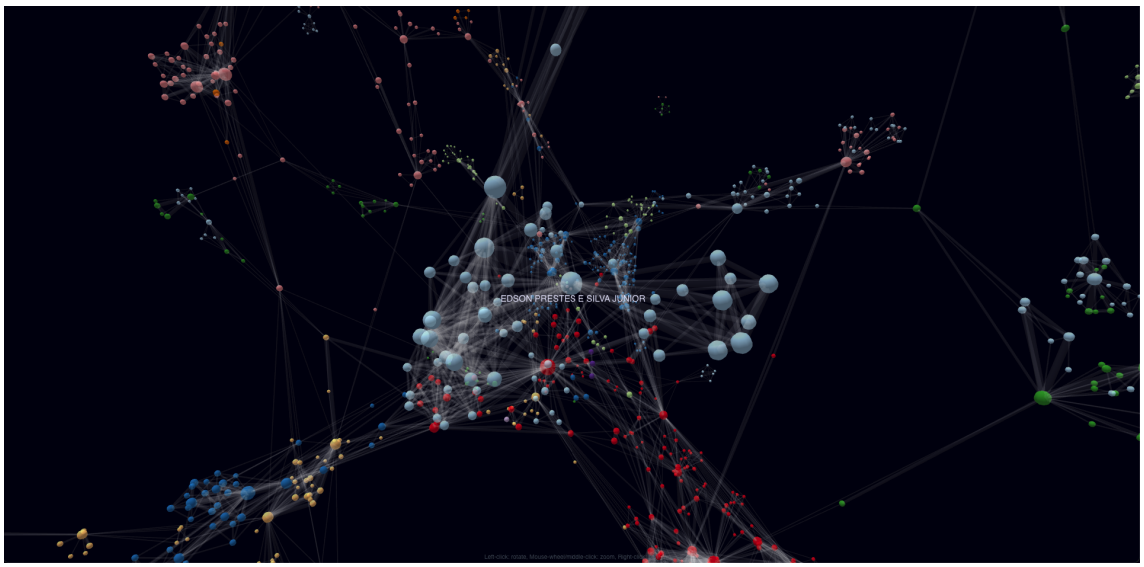


Figura 2.8 – Visualização de um grafo de co-autorias do PPGC UFRGS (WICKBOLDT, 2019) utilizando a biblioteca 3D Force Graph (ASTURIANO, 2017).

A ferramenta também permite a inserção e remoção dinâmica de nós e arestas,

o que possibilita a implementação de uma solução de visualização adaptativa através do carregamento de informações sob-demanda em visualizações com grande quantidade de dados.

2.3.2 Neo4J (NEO4J, 2022a)

Neo4j é um sistema de gerenciamento de banco de dados gráfico nativo desenvolvido especialmente para o processamento e armazenamento de dados altamente relacionados (NEO4J, 2022b). Esta arquitetura voltada a grafos possibilita consultas altamente eficientes a entidades e relacionamentos. O sistema também possui *drivers* para as principais linguagens de programação, permitindo integração direta entre a aplicação e a base de dados e possibilitando a realização de consultas de dados sob demanda pela aplicação de visualização.

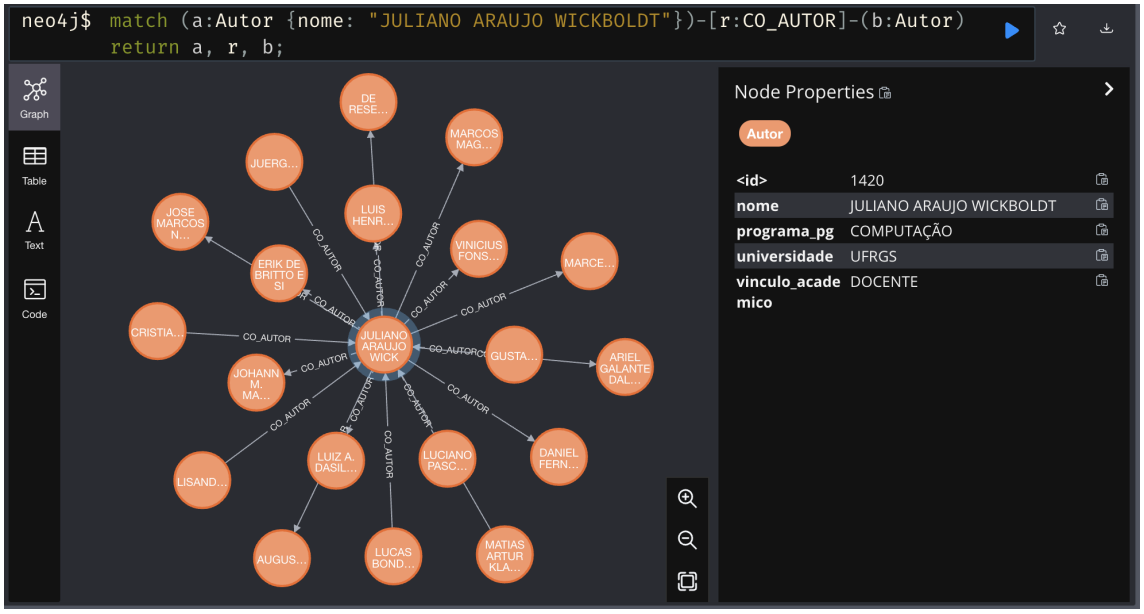


Figura 2.9 – Consulta efetuada na base de dados Neo4j requisitando todos os nós que possuem o relacionamento CO_AUTOR com o Autor de nome JULIANO ARAUJO WICKBOLDT dentro de uma rede composta por autores e produções acadêmicas dos anos de 2017 a 2019 (CAPES, 2022a). A consulta em questão levou 5ms para ser concluída.

Para avaliar a viabilidade do uso desta base de dados na composição da solução proposta neste trabalho, foram realizados testes envolvendo diferentes volumes de dados. Durante a realização dos testes, foi possível observar que o tempo de consulta para uma consulta simples (Figura 2.9) não apresenta grandes variações ao aumentar o tamanho da rede armazenada no banco de dados, permanecendo abaixo de 10ms para todas os *da-*

tasets testados e aparentando variar proporcionalmente ao número de nós retornados na consulta. O tamanho da base de dados também parece se manter na ordem de dezenas de *megabytes* para o volume de dados testado, o que não aparenta ser motivo de preocupações haja vista que os dispositivos de armazenamento presentes nos computadores atuais têm capacidade de armazenamento maior em ordens de magnitude. O terceiro aspecto avaliado, e que se mostrou um potencial fator limitante da escala da rede de colaborações, foi o tempo de importação do conjunto de dados para a banco de dados Neo4j.

A Tabela 2.1 juntamente com o gráfico presente na Figura 2.10 apresentam os dados a respeito do tempo de importação mensurado para diferentes conjuntos de dados.

Dataset	Produções Intelectuais	Autores	Tempo para importação (s)	Produções/seg
CAPES Anais UFRGS 2019	4.948	8.190	56	88,4
CAPES Anais + Livros UFRGS 2019	7.300	17.196	158	46,2
CAPES Anais + Livros UFRGS 2018 - 2019	14.676	17.471	495	29,6
CAPES Anais + Livros UFRGS 2017 - 2019	22.753	23.926	1021	22,3

Tabela 2.1 – Tempo de importação e taxa de importação de dados de autoria de produções intelectuais para o banco de dados Neo4J para diferentes volumes de dados

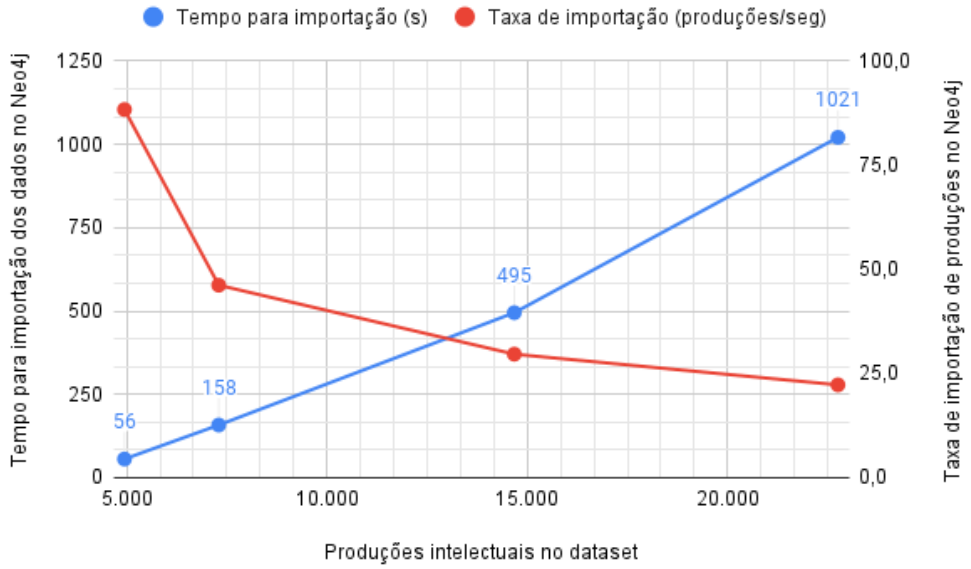


Figura 2.10 – Variação do tempo de importação e da taxa de produções importadas por segundo para *datasets* de diferentes tamanhos.

Os testes consistiram na importação de dados de dois *datasets* da CAPES (CAPES, 2022a) referentes aos anos de 2017 a 2019: Produções acadêmicas e Autores de produções acadêmicas. O processo de importação se deu através das seguintes etapas:

1. Concatenação de arquivos .csv fornecidos pela CAPES com dados de diferentes anos para um arquivo .csv único.
2. *Parsing* do arquivo .csv resultante para cada *dataset* com sanitização dos campos,

filtragem dos dados de interesse (ano, universidade) e geração de arquivo .csv contendo apenas os campos de interesse para o conjunto de dados em questão.

3. Importação dos dados referentes a autores de produções acadêmicas no Neo4j com criação de entidades dos tipos *Autor* e *Produção* (nesta etapa contendo apenas o identificador numérico da produção acadêmica referenciado no *dataset* de autores), bem como a criação de um relacionamento do tipo *AUTOR* entre o autor e a produção. Segue abaixo a *query* Neo4j referente à etapa:

```
LOAD CSV WITH HEADERS
FROM 'file:///autores.csv' AS autor
MERGE (a:Autor {nome: autor.nome})
SET a.universidade = autor.universidade,
    a.programa_pg = autor.nome_programa_pg,
    a.vinculo_academico = autor.vinculo_acad
MERGE (p:Producao {id: autor.id_producao})
MERGE (a)-[:AUTOR]->(p);
```

4. Importação dos dados referentes a produções acadêmicas, preenchendo as entidades do tipo *Produção* criadas na etapa com detalhes a respeito do trabalho ou criando uma nova entidade pra cada produção acadêmica não referenciada no conjunto de dados de autores. Segue abaixo a *query* Neo4j referente à etapa:

```
LOAD CSV WITH HEADERS
FROM 'file:///producoes.csv' AS producao
MERGE (p:Producao {id: producao.id})
SET p.titulo = producao.titulo,
    p.ano = producao.ano,
    p.programa_pg = producao.nome_programa_pg,
    p.universidade = producao.universidade,
    p.tipo = producao.tipo,
    p.linha = producao.linha;
```

5. Consulta todas as produções acadêmicas existentes no Neo4j e cria relações do tipo *CO-AUTOR* entre autores de uma mesma produção. Segue abaixo a *query* Neo4j referente à etapa:

```
MATCH (p:Producao)
WHERE size((p)<-[:AUTOR]-()) > 1
WITH [(p)<-[:AUTOR]-(a) | a] as coAutores
UNWIND coAutores as first
UNWIND coAutores as second
WITH first, second
WHERE id(first) < id(second)
MERGE (first)-[:CO_AUTOR]-(second);
```

Observando o gráfico da Figura 2.10, é possível observar que o tempo de importação aumenta com o tamanho do *dataset* enquanto a taxa de importação diminui, o que indica um crescimento não linear do tempo de importação para conjuntos de dados com número maior de elementos. Considerando que este trabalho almeja criar uma rede de colaborações acadêmicas de escala nacional, abrangendo todas as universidades brasileiras cadastradas na base de dados abertos da CAPES (CAPES, 2022a), com informações completas do último quadriênio publicado e incluindo produções publicadas em periódicos, anais, jornais e revistas, estima-se que o volume de dados final a ser importado esteja na casa de milhões de pesquisadores e produções intelectuais. Esses números se mostram 2 ordens de magnitude maiores comparados ao volume de dados utilizados nos testes apresentados acima, de forma que a importação dos dados na base de dados não parece escalável se feita da forma descrita nesta seção.

Para a utilização da base de dados Neo4J na solução final deste trabalho, foi necessário otimizar o processo de importação de forma a torná-lo mais eficiente a fim de importar um grande volume de dados em tempo hábil. Mais detalhes sobre as medidas tomadas para a otimização desse processo, bem como os novos resultados obtidos podem ser encontrados na seção 3.3.

3 GERAÇÃO DA REDE DE COLABORAÇÃO ACADÊMICA

A solução implementada por este trabalho pode ser dividida em duas partes. A primeira parte consiste na concepção de uma rede de colaborações acadêmicas gerada a partir de dados disponibilizados no portal Dados Abertos CAPES (CAPES, 2022a). A geração dessa rede é composta por 5 etapas: (1) seleção dos conjuntos de dados de interesse, (2) pré-processamento, (3) agrupamento, (4) pós-processamento e (5) importação no banco de dados de grafos. O resultado é uma rede de colaboração multi nível composta por 532 universidades brasileiras, 4.685 programas de pós-graduação, 1.275.852 autores, 1.708.666 produções acadêmicas e 14.883.507 relações de co-autoria de trabalhos acadêmicos. A segunda parte diz respeito ao desenvolvimento de uma aplicação de visualização desta rede e será detalhada no capítulo 4.

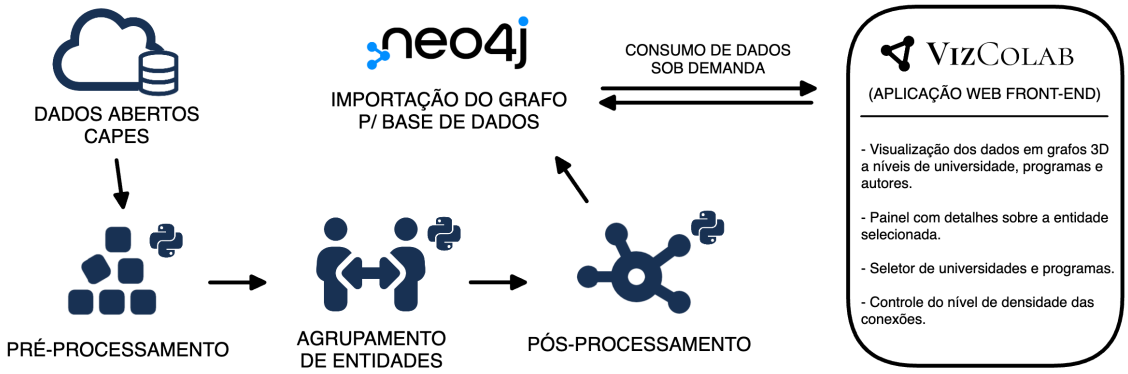


Figura 3.1 – Esquemático ilustrando a arquitetura da solução implementada

Este capítulo descreve os recursos e os métodos utilizados para a geração da rede de colaboração acadêmica de escala nacional concebida neste trabalho. O capitulo está subdividido em três seções, tratando do processo de obtenção de dados, processamento de dados e importação e organização do grafo de co-autorias no banco de dados.

3.1 Obtenção de Dados

Este trabalho se propõe a reunir dados a respeito de universidades, programas de pós-graduação, trabalhos acadêmicos e pesquisadores com a finalidade de gerar uma rede de colaborações acadêmica de nível nacional. O conjunto de dados resultante deve refletir da forma mais acurada possível o estado dos relacionamentos de colaborações acadêmicas brasileiro. Com esse objetivo, o conjunto de dados escolhido como base para a geração do grafo de co-autorias foi o conjunto de dados abertos (CAPES, 2022a) disponibilizados

pela CAPES.

A CAPES é uma fundação vinculada ao Ministério da Educação do Brasil que atua na expansão e consolidação da pós-graduação *stricto sensu* em todos os estados brasileiros. A entidade mantém e disponibiliza bases de dados abertas com informações sobre diversos aspectos dos programas de pós-graduação brasileiros.

Dentre os diversos conjuntos de dados disponíveis na plataforma, quatro *datasets* foram selecionados por conterem dados de interesse à aplicação final. A seguir, encontra-se uma lista com os conjuntos de dados utilizados, bem como os pontos de dados coletados de cada conjunto:

1. Autor da produção intelectual de programas de pós-graduação *stricto sensu* no Brasil (CAPES, 2022b)
    - **Nome do autor:** Nome completo do autor da produção acadêmica.
    - **Nome ABNT do autor:** Nome do autor da produção no formato de abreviação ABNT. (Ex.: Oliveira, V. A.)
    - **Tipo do autor:** Tipo de vínculo do autor com o programa de pós graduação. (Ex.: docente, discente, participante externo, ...)
    - **Categoria docente:** Caso o autor seja do tipo docente, descreve sua categoria. (Ex.: colaborador, permanente, ...)
    - **Nível discente:** Caso o autor seja do tipo discente, descreve seu nível. (Ex.: bacharelado, mestrado, doutorado, ...)
    - **Código do programa IES:** Código do programa de pós-graduação na CAPES.
    - **Nome do programa IES:** Nome do programa de pós-graduação na CAPES.
    - **Área do conhecimento:** Nome da área de conhecimento (Ex.: administração, odontologia, sociologia, ...)
    - **Sigla da entidade de ensino:** Sigla da instituição de ensino do programa de pós-graduação. Se programa em rede, representa a sigla da instituição principal (Ex.: UFRGS, UFPEL, PUCRS, ...)
    - **Id pessoa discente:** Número de identificação do discente enquanto pessoa na base de dados da CAPES.
    - **Id pessoa docente:** Número de identificação do docente enquanto pessoa na base de dados da CAPES.
    - **Id participante externo:** Número de identificação do participante externo

enquanto pessoa na base de dados da CAPES.

- **Id pós-doutorando:** Número de identificação do pós-doutorando enquanto pessoa na base de dados da CAPES.
- **Id egresso:** Número de identificação do egresso enquanto pessoa na base de dados da CAPES.
- **Id produção intelectual:** Identificação do produto no ano base e no programa na base de dados da CAPES.

2. Produção intelectual de pós-graduação *stricto sensu* no Brasil (CAPES, 2022d)

    - **Nome da produção:** Título do produto
    - **Tipo da produção:** Tipo de produção na base de dados da CAPES. (Ex.: técnica, bibliográfica, ...)
    - **Subtipo da produção:** Subtipo de produção na base de dados da CAPES. (Ex.: livro, periódico, jornal, ...)
    - **Ano base:** Ano de referência cadastrado no sistema Coleta CAPES[1].
    - **Sigla da entidade de ensino:** Sigla da instituição de ensino do programa de pós-graduação. Se programa em rede, representa a sigla da instituição principal (Ex.: UFRGS, UFPEL, PUCRS, ...)
    - **Nome do programa IES:** Nome do programa de pós-graduação na CAPES.
    - **Área de concentração:** Área de concentração do programa a qual está vinculada a produção. (Ex.: clínica odontológica, engenharia e ciência dos materiais, ...)
    - **Linha de pesquisa:** Linha de pesquisa do programa a qual está vinculada a produção. (Ex.: energias renováveis e não renováveis, diversidade biológica e ecológica, ...)
    - **Nome do projeto:** Projeto de pesquisa do programa a qual está vinculada a produção.
    - **Id produção intelectual:** Identificação do produto no ano base e no programa na base de dados da CAPES.

3. Programas da pós-Graduação *stricto sensu* no Brasil (CAPES, 2022e)

    - **Sigla da entidade de ensino:** Sigla da instituição de ensino do programa de pós-graduação. Se programa em rede, representa a sigla da instituição

[1] https://sucupira.capes.gov.br/sucupira/

principal (Ex.: UFRGS, UFPEL, PUCRS, ...)

- **Nome do programa IES:** Nome do programa de pós-graduação na CAPES.
- **Código do programa IES:** Código do programa de pós-graduação na CAPES.
- **Grande área do conhecimento:** Grande área do conhecimento do programa de pós-graduação. (Ex.: ciências humanas, ciências agrárias, ...)
- **Área do conhecimento:** Área de conhecimento do programa de pós-graduação (Ex.: administração, geografia, letras, ...)
- **Sub-área do conhecimento:** Sub-área do conhecimento do programa de pós-graduação. (Ex.: sociais e humanidades, literatura comparada, ...)
- **Especialidade:** Especialidade do conhecimento do programa de pós-graduação. (Ex.: doenças infecciosas e parasitárias, química dos produtos naturais, ...)
- **Área de avaliação:** Área de avaliação do programa de pós-graduação. (Ex.: psicologia, ensino, ciências ambientais, ...)

4. Cursos da pós-graduação *stricto sensu* no Brasil (CAPES, 2022c)

    - **Código entidade CAPES:** Código da instituição de ensino superior na CAPES.
    - **Código entidade e-MEC:** Código da instituição de ensino superior no sistema e-MEC.
    - **Nome entidade de ensino:** Nome da instituição de ensino
    - **Status jurídico:** Classificação do status jurídico. (Ex.: federal, estadual, particular, ...)
    - **Dependência administrativa:** Descrição da dependência administrativa. (Ex.: pública, privada, ...)
    - **Nome da região:** Nome da região da instituição de ensino. (Ex.: sul, sudeste, norte, ...)
    - **Sigla da UF:** Sigla da unidade federativa da instituição de ensino. (Ex.: RS, SC, SP, ...)
    - **Município do programa IES:** Nome do município do programa da IES. (Ex.: Porto Alegre, Pelotas, ...)

A CAPES disponibiliza os seus conjuntos de dados de forma quadrienal. Para a geração da rede de colaborações aqui apresentada, foram utilizados os dados mais re-

centes disponíveis na plataforma, que no momento do encerramento deste trabalho, são referentes ao quadriênio de 2017 a 2020.

3.2 Processamento de Dados

Os dados obtidos no portal de dados abertos da CAPES (CAPES, 2022a) são disponibilizados em arquivos no formato *.csv*. Cada um dos *datasets* utilizados é composto por um conjunto de arquivos *.csv*, dentre os quais os dados são divididos por tipo e por ano. O objetivo desta etapa de processamento de dados é transformar esse conjunto de arquivos obtidos — contendo dados sobre autores, produções intelectuais, programas de pós-graduação e cursos de ensino superior — em um grafo de co-autorias capaz de sintetizar o estado das colaborações acadêmicas no país, incluindo dados sobre colaborações entre autores, programas e universidades, produções acadêmicas e as respectivas relações de autoria com seus autores, além de informações detalhadas a respeito de cada uma dessas entidades (autores, programas, universidades e produções).

As transformações de dados necessárias foram realizadas utilizando a linguagem de programação Python (PYTHON, 2022), com apoio de *notebooks* Jupyter (JUPYTER, 2022). Para facilitar a manipulação dos dados, foi utilizada a biblioteca Pandas (PANDAS, 2022). As etapas de processamento de dados são realizadas individualmente para cada um dos *datasets* utilizados no trabalho (autores, produções intelectuais, programas e cursos). Embora existam diferenças entre as ações desempenhadas sobre cada um dos conjuntos de dados, o processamento pode ser subdividido, de modo geral, em três etapas: pré-processamento, agrupamento de entidades e pós-processamento. A seguir, serão detalhas as ações realizadas sobre cada conjunto de dados dentro das etapas enumeradas acima:

3.2.1 Conjunto de Dados de Autor da Produção Intelectual

O *dataset* de autores de produções intelectuais utilizado no trabalho é composto por um conjunto de 10 arquivos no formato *.csv*, sendo eles:

- autores-anais-2017.csv (838.825 registros, 164,6 MB)
- autores-anais-2018.csv (792.341 registros, 155,3 MB)
- autores-anais-2019.csv (802.192 registros, 156,6 MB)

- autores-anais-2020.csv (406.248 registros, 79 MB)
- autores-periodicos-2017.csv (1.105.052 registros, 222,6 MB)
- autores-periodicos-2018.csv (1.192.718 registros, 239,9 MB)
- autores-periodicos-2019.csv (1.298.203 registros, 260,6 MB)
- autores-periodicos-2020.csv (1.547.249 registros, 311,1 MB)
- autores-livros-2017-2020.csv (990.699 registros, 192,6 MB)
- autores-jornais-revistas-2017-2020.csv (140.554 registros, 26,6 MB)

Cada um desses arquivos contém uma lista de entradas correspondendo a um par autor – produção do tipo e ano correspondente ao arquivo. Isto é, para cada autor de cada produção intelectual cadastrada no sistema CAPES, existe uma linha no *dataset* de autores especificando 30 propriedades a respeito do papel do autor na produção intelectual em questão.

PRÉ-PROCESSAMENTO

O pré-processamento desses dados se inicia pela concatenação dos diversos arquivos em uma tabela única de autorias. Em seguida, as colunas da tabela são filtradas, selecionando apenas as 15 colunas de interesse descritas da seção 3.1 dentre as 30 colunas disponíveis no *dataset*.

Observando os pontos de dados selecionados para o conjunto de autores na seção 3.1, é possível notar que existem cinco campos de identificadores (*Id*) distintos para cada entrada na tabela, sendo eles:

- Id pessoa discente
- Id pessoa docente
- Id participante externo
- Id pós-doutorando
- Id egresso

Isso ocorre pois a CAPES designa identificadores distintos para cada tipo de autor. Como cada entrada da tabela é categoriza por um único tipo, conforme o valor do campo *Tipo do autor*, os diversos identificadores se comportam como colunas de uma matriz esparsa. Isso significa que para cada entrada da tabela, apenas um dos valores de *Id* listados acima estará preenchido. É de interesse desta aplicação, para facilitar tratamentos futuros, que cada entrada possua um único identificador. Por conta disso, a próxima etapa do pré-processamento consiste em substituir essas cinco colunas de identificador por uma

única coluna denominada *ID*, sendo esta preenchida com o identificador correspondente ao tipo da entrada determinada por sua propriedade *Tipo do autor*.

Fazendo uma análise rápida do conjunto de dados, é possível perceber que um mesmo autor pode ter diferentes ocorrências na tabela. Entre essas ocorrências, pode haver divergência entre os valores da propriedade *Nome do autor*. Algumas formas em que essa situação pode se manifestar são através de divergências nas ocorrências de acentos e sinais de pontuação entre as entradas do autor, existência de espaçamentos adicionais, caracteres especiais, etc... Existem ainda casos onde o nome do autor é cadastrado no formato "Sobrenome, Nome". Essas divergências se tornam um problema na próxima fase do processamento de dados (*Agrupamento*). Por isso, o pré-processamento inclui uma etapa de normalização de nomes, onde são realizados os seguintes procedimentos:

- Inversão do nome do autor no caso da presença do caractere ","(vírgula). Ex.: Um nome cadastrado como "da Silva, João", será convertido para "João da Silva"
- Remoção de acentos
- Remoção de caracteres especiais
- Remoção de espaçamentos extras (espaços no início ou fim do nome, espaços duplos, *tabs*, quebras de linha, ...)

Após a normalização, a probabilidade de um mesmo autor ter nomes distintos em diferentes entradas do conjunto de dados é reduzida.

AGRUPAMENTO

Um mesmo autor pode aparecer repetidas vezes na tabela de dados. Isso se deve a natureza do conjunto de dados, que inclui uma nova entrada para cada produção autorada por um determinado autor. Para a presente aplicação, é fundamental que as diversas entradas correspondentes a um mesmo autor possam ser agrupadas de maneira adequada, de forma que cada autor seja representado por uma única entidade no grafo de colaborações resultante. A etapa de agrupamento de autores é realizada em três níveis, utilizando três técnicas distintas, sendo elas: agrupamento por identificador, agrupamento por nome e agrupamento por pontuação.

Cada entrada na tabela possui valores para cada uma das colunas que estão sendo trabalhadas. Ao realizar o *merge* de dois autores, é possível que ambos possuam valores distintos para diferentes colunas, como *Nome do autor*, *Tipo do autor*, *Sigla da entidade de ensino*, etc. Ao realizar o agrupamento, é preciso decidir como lidar com esses diferentes valores para cada coluna. Neste trabalho, esses conflitos de valores foram tratados de

quatro formas distintas, definindo quatro funções de agregação que podem ser aplicadas individualmente para cada uma das colunas, sendo elas:

- **Valor mínimo:** Seleciona o valor mínimo entre as diferentes ocorrências.
- **Primeiro valor:** Seleciona o valor correspondente a primeira ocorrência.
- **Lista de valores:** Cria uma lista contendo todos os diferentes valores ocorridos.
- **Contagem de valores:** Cria uma estrutura de dicionário contento pares chave-valor, onde a chave corresponde a um determinado valor ocorrido e o valor corresponde a quantidade de ocorrências desse valor. (Ex.: {'UFRGS': 18, 'UFPEL': 2, 'PUCRS': 3})

Dadas essas funções de agregação, o mapeamento das funções para cada coluna da nossa tabela de dados foi definida da seguinte forma:

- **Id:** Valor mínimo
- **Nome do autor:** Contagem de valores
- **Nome ABNT do autor:** Contagem de valores
- **Tipo do autor:** Contagem de valores
- **Categoria docente:** Contagem de valores
- **Nível discente:** Contagem de valores
- **Código do programa IES:** Contagem de valores
- **Nome do programa IES:** Contagem de valores
- **Área do conhecimento:** Contagem de valores
- **Sigla da entidade de ensino:** Contagem de valores
- **Id da produção intelectual:** Lista de valores

É possível notar que, com exceção do *Id* e do *Id da produção intelectual*, todas as demais colunas utilizam a função de agregação *Contagem de valores*. A escolha dessa função para o agrupamento da maioria das propriedades se dá pelo fato de esta ser a que mantém a maior quantidade de informações a respeito dos dados originais do *dataset* ao longo das diversas etapas de agrupamento, permitindo que ao final do processo, decisões sobre quais serão as propriedades finais de cada autor possam ser tomadas de forma mais precisa. Para a coluna de *Id*, a função de valor mínimo foi escolhido pois a existência de múltiplos identificadores por autor não nos agrega informações relevantes, de forma que apenas o identificador de menor valor numérico é mantido. Na coluna de *Id da produção intelectual*, por outro lado, é de suma importância que seja mantida uma lista de todas

as produções intelectuais de autoria do autor em questão, justificando assim a escolha da função de agregação *Lista de valores*.

O primeiro método aplicado foi o agrupamento por identificador *Id*. Em um cenário ideal, um mesmo autor deveria ter o mesmo identificador em todas as suas ocorrências no conjunto de dados, de forma que um agrupamento por *Id* seria suficiente para reunir as entradas correspondentes a um mesmo autor. Analisando o *dataset*, entretanto, percebe-se que embora esse seja o caso para uma quantidade significativa dos dados, existem duas situações onde essa técnica não se mostra efetiva. Em primeiro lugar, nota-se que existem autores sem identificadores na base de dados, isto é, todos os possíveis campos de *Id* estão vazios. Em segundo lugar, se trata de autores cadastrados múltiplas vezes no conjunto de dados com identificadores distintos. Para ambos os casos, é preciso utilizar diferentes métodos de agrupamento, que serão apresentados em um segundo momento.

Logo após a importação dos conjuntos de dados, antes da realização do *merge* por identificadores, obteve-se um total de 9.114.081 entradas na tabela de dados. Após a realização do agrupamento por *Ids*, o número de entradas é reduzido para 2.081.451. Isso significa que 77,2% das entradas da tabela possuíam identificadores com mais de uma ocorrência no conjunto de dados e foram agrupadas.

Analisando o conjunto de dados resultante, foi possível observar que 1.283.808 dos 2.081.451 autores restantes (61,7%) não possuem valores na coluna *Id*. É possível perceber que, enquanto o método de agrupamento por identificador foi bastante efetivo entre o conjunto de autores com *Id*s corretamente cadastrados, uma parcela significativa da lista de autores resultante não possui identificadores e, portanto, não foi afetada pelo método. É razoável assumir que muitos desses autores ainda possuem múltiplas ocorrências dentro do conjunto de dados, de forma que uma maneira alternativa de agrupamento se faz necessária.

Na tentativa de amenizar este problema e obter um conjunto de dados mais consistente, foi aplicado um segundo método de agrupamento baseado no nome do autor. O método consiste em agrupar todos os autores de nome exatamente igual. Embora exista a possibilidade de que casos falso positivos levem a situação onde autores distintos que compartilham o mesmo nome completo sejam agrupados indevidamente, observou-se que esses casos são bastante raros, de forma que a alta taxa de acertos nos agrupamento levou ao prosseguimento da aplicação do método. Vale lembrar que na etapa de pré-processamento (detalhada acima) os nomes completos dos autores foram normalizados. Isso faz com que muitos dos erros de comparação ocasionados por diferenças de grafia,

acentuação, espaçamentos ou uso de caracteres especiais sejam contornados, resultando em uma maior taxa de sucesso.

Após a aplicação dessa segunda etapa de agrupamentos, o número entradas foi reduzido para 1.275.852, uma redução de 38,7% em relação à etapa anterior e 86% em relação ao conjunto de dados inicial. Em uma análise do conjunto de dados obtido após as duas etapas de agrupamento de autores, é possível perceber que o resultado do agrupamento parece satisfatório. Nessa etapa, a grande maioria dos autores se encontra completamente agrupadas, sem múltiplas entradas na tabela. É necessária uma análise mais atenta para perceber que existem algumas exceções. Essas se dão por aqueles autores que foram cadastrados na base de dados sem um identificador ao mesmo tempo que receberam nomes distintos entre algumas de suas ocorrências.

O desenvolvimento de uma solução algorítmica para este problema é difícil. Embora existam poucas ocorrências de pessoas com nomes completos exatamente iguais, ocorrências de pessoas com nomes semelhantes são muito mais comuns, de forma que inferir quais autores dentro de um conjunto de larga escala são semelhantes o suficiente a ponto de serem considerados a mesma pessoa pelo algoritmo se torna uma tarefa significativamente mais complexa e computacionalmente custosa.

O terceiro método de agrupamento desenvolvido neste trabalho é uma tentativa de solucionar este problema. A abordagem utilizada consiste em utilizar um sistema de pontuação para determinar se um par de entradas na tabela de autores é semelhante o suficiente para que ambas as entradas sejam consideradas a mesma pessoa, sendo portanto agrupadas entre si. Ao comparar duas entradas da tabela, o par de entradas recebe pontos ao atingir determinados critérios de semelhança. Os pontos são atribuídos da seguinte forma:

- O par de autores recebe **2 pontos** caso possuam **primeiro e último nomes iguais**. Por exemplo, um par de autores com nomes "João Lucas da Silva"e "João da Silva"satisfazem esse critério, recebendo a pontuação. Esse critério de pontuação foi implementado após observação de que a ocultação de nomes do meio são motivo frequente de divergência de nomes. Esse é o critério de pontuação de maior peso.
- O par de autores recebe **1 ponto** caso possuam **valores iguais do campo *Nome ABNT do autor***. Vale lembrar que após os agrupamentos iniciais, essa coluna da tabela armazena uma contagem de valores, de forma os grupos de valores de cada autor precisam ser comparados par a par. Uma única correspondência é suficiente para

a obtenção da pontuação. Por exemplo, um autor de *Nome ABNT do autor* igual a `{"da Silva, J. L.": 3, "da Silva, J.": 2}` pontuará ao ser comparado com outro autor com o valor do campo igual a `{"da Silva, J.": 1}`. Esse critério de pontuação foi implementado após observação de que a abreviação de nomes do meio são motivo frequente de divergência de nomes.

- O par de autores recebe **1 ponto** caso possuam valores iguais do campo *Sigla da entidade de ensino*. Esta coluna também é armazenada no formato *contagem de valores*, de modo que a comparação se dá da mesma forma que a propriedade acima. Esse critério de pontuação foi implementado assumindo que autores de nomes semelhantes pertencentes a uma mesma instituição de ensino tem maior probabilidade de se tratarem do mesmo indivíduo.
- O par de autores recebe **1 ponto** caso possuam valores iguais do campo *Tipo de autor* (docente, discente, etc). Esta coluna também é armazenada no formato *contagem de valores*, de modo que a comparação se dá da mesma forma que a propriedade acima. Esse critério de pontuação foi implementado assumindo que autores de nomes semelhantes de um mesmo tipo tem maior probabilidade de se tratarem do mesmo indivíduo.
- Por fim, o par de autores recebe **1 ponto** caso possuam valores iguais do campo *Nome do programa IES*. Esta coluna também é armazenada no formato *contagem de valores*, de modo que a comparação se dá da mesma forma que as propriedades acima. Esse critério de pontuação foi implementado assumindo que autores de nomes semelhantes pertencentes a um mesmo programa de pós graduação tem maior probabilidade de se tratarem do mesmo indivíduo.

Uma função de comparação foi implementada de forma a retornar uma pontuação que varia entre 0 e 6 pontos. Após testes com diferentes valores de pontuação de corte para a decisão de que um par de autores é semelhante o suficiente para serem considerados a mesma pessoa, escolheu-se a pontuação de mínima de 5 pontos para que o resultado da comparação leve os autores a serem agrupados entre si com elevada probabilidade de acerto.

Embora a heurística de pontuação desenvolvida seja consideravelmente simples, a sua aplicação sobre o extensivo conjunto de dados não é trivial. Enquanto os agrupamentos anteriores se baseiam na correspondência exata de valores de uma única coluna da tabela, permitindo a utilização de técnicas como *hash tables* para a identificação eficiente de correspondências, o sistema de pontuação exige comparações mais complexas,

Etapa	Descrição da etapa	Entradas resultantes	Agrupamento	Tempo de processamento
0	Dataset original	9.114.081	0%	-
1	Agrupamento por Id	2..081.451	77,2%	9m 44s
2	Agrupamento por Nome	1.275.852	86%	10m 27s
3	Agrupamento por Pontuação	1.212.059	86,7%	5 dias e 3 horas (estimado)

Tabela 3.1 – Resultados de cada uma das etapas de agrupamento de autores

envolvendo múltiplas propriedades de cada item da tabela, demandando, portanto, comparações par a par extensivas. Testes iniciais que realizam a comparação de cada um dos autores com todos os demais se mostraram computacionalmente proibitivos dadas as proporções da base de dados, com estimativas na casa de dezenas de dias de processamento utilizando o hardware disponível[2].

Com a finalidade de reduzir o tempo de processamento, viabilizando a execução desta etapa de processamento no hardware disponível, foram tomadas as medidas de otimização detalhadas a seguir:

- Paralelização do processamento: Por padrão, o processamento do código *Python* no *notebook Jupyter* é realizado em uma única *thread* do processador. Esta otimização utiliza a biblioteca *joblib* para permitir a execução *multithread* do algoritmo.
- Interrupção prematura da comparação: O sistema de pontos descrito acima requer que a comparação entre um par de autores retorna 5 pontos ou mais para que ambos sejam considerados o mesmo autor. Essa otimização consiste em abortar a comparação uma vez que o par de autores não possui mais possibilidade matemática de atingir os pontos necessários, evitando as demais comparações.
- Uso de dicionários: Inicialmente a iteração entre autores era feita sobre uma estrutura de dados do tipo *pandas dataframe*. Essa otimização consiste em converter os dados para uma estrutura de dicionário da linguagem *Python* (*dict*), resultando em uma computação mais eficiente.

Após a implementação dessas otimizações, o tempo de processamento da última etapa foi reduzido para 5 dias e 3 horas. Embora seja um tempo elevado, foi possível reduzir o tempo de processamento inicial desta etapa em cerca de 6 vezes.

PÓS-PROCESSAMENTO

Ao final da etapa de agrupamento, obteve-se uma tabela com o conjunto final de

[2]Todos os tempos de processamento descritos neste trabalho se referem a execução em um MacBook Pro 2021, Chip M1 Pro, 16GB RAM, 512GB armazenamento

autores que integram a rede de colaborações acadêmicas. É preciso lembrar, entretanto, que algumas das colunas da tabela ainda são compostas por estruturas de agregação (como *listas de valores* e *contagens de valores* contendo múltiplos valores para cada propriedade dos autores. Esta etapa de pós-processamento implementa heurísticas de preferência, com o objetivo de decidir um único valor final para cada propriedade de um determinado autor. As heurísticas aplicadas são detalhadas a seguir para cada uma das colunas:

- **Tipo do autor:** O valor dessa coluna é escolhido com base em uma *lista de preferências*.Os valores possíveis para este campo são definidos em ordem decrescente de preferência: `['DOCENTE', 'EGRESSO', 'PÓS-DOC', 'DISCENTE', 'PARTICIPANTE EXTERNO']`. O valor final do campo será aquele de maior preferência dentro do conjunto de ocorrências armazenado para o autor. Essa estratégia se baseia na suposição de que autores acadêmicos tendem a escalar posições dentro de uma instituição de ensino. Assume-se, por exemplo, que para um autor com publicações com tipo 'DISCENTE' e 'DOCENTE', é razoável assumir que seu tipo mais recente seja 'DOCENTE'.
- **Categoria docente:** De forma similar, esta coluna também implementa uma *lista de preferências*, definida da seguinte forma: `['PERMANENTE', 'COLABORADOR', 'VISITANTE']`. Essa estratégia se baseia na suposição de que docentes escalam posições dentro de uma instituição de ensino, assumindo, por exemplo, que um docente permanente não voltará a ser um docente visitante.
- **Nível discente:** Essa propriedade implementa a seguinte *lista de preferências*: `['DOUTORADO', 'DOUTORADO PROFISSIONAL', 'MESTRADO', 'MESTRADO PROFISSIONAL', 'BACHARELADO']`.
- As demais propriedades com múltiplos valores (**'Nome do autor', 'Nome ABNT do autor', 'Código do programa IES', 'Nome do programa IES', 'Área do conhecimento', 'Sigla da entidade de ensino'**) são armazenadas no formato *Contagem de valores*. Sobre essas propriedades, é escolhido como valor final aquele que aparece com maior frequência dentre as produções acadêmicas de um determinado autor.

Após a seleção das propriedades finais, uma coluna é adicionada à tabela: *Contagem de produções*. Essa coluna armazena a contagem de produções intelectuais de cada autor e tem o intuito de ser utilizada para facilitar a contabilização de produções acadêmicas produzidas por autores, programas de pós-graduação e universidades em etapas

futuras do trabalho. Ao final do processamento dos dados referentes a autores de produções intelectuais, foi produzido um arquivo de saída:

- **processed_authors.csv:** Contém o conjunto final de autores de produções intelectuais processados nesta etapa.

3.2.2 Conjunto de Dados de Produções Intelectuais

O *dataset* de produções intelectuais utilizado no trabalho é composto por um conjunto de 4 arquivos no formato *.csv*, sendo eles:

- producoes-anais-2017-2020.csv (821.672 registros, 370,2 MB)
- producoes-periodicos-2017-2020.csv (1.162.324 registros, 529,3 MB)
- producoes-livros-2017-2020.csv (424.489 registros, 179,3 MB)
- producoes-jornais-revistas-2017-2020.csv (97.446 registros, 38,7 MB)

Cada um desses arquivos contém uma lista de entradas correspondendo a produções intelectuais cadastradas no sistema CAPES de um determinado tipo (anais, periódicos, livros, jornais e revistas). Isto é, para cada produção intelectual cadastrada no sistema CAPES, existe uma linha no *dataset* de produções especificando 30 propriedades a respeito da produção intelectual em questão.

PRÉ-PROCESSAMENTO

O pré-processamento desses dados se inicia pela concatenação dos diversos arquivos em uma tabela única de produções intelectuais. Em seguida, as colunas da tabela são filtradas, selecionando apenas as 10 colunas de interesse descritas da seção 3.1 dentre as 27 colunas disponíveis no *dataset* original.

Assim como é feito com os autores, o conjunto de dados de produções intelectuais também passa por um processo de normalização de títulos. A normalização é feita com o intuito de remover caracteres e espaçamentos inválidos que possam prejudicar a apresentação da informação, bem como aprimorar a etapa futura de agrupamento de produções intelectuais de mesmo título. Neste caso, o processo de normalização consiste nos seguintes procedimentos:

- Remoção de aspas simples e duplas do início e final do título.
- Remoção de espaçamentos extras (espaços no início ou fim do nome, espaços duplos, *tabs*, quebras de linha, ...)

AGRUPAMENTO

O conjunto de dados de produções intelectuais, ao contrário do conjunto de autores, possui apenas uma linha para cada produção intelectual cadastrada no sistema da CAPES. O intuito dessa etapa de agrupamento, portanto, passa a ser exclusivamente o de detectar produções intelectuais que, por alguma razão, foram cadastradas múltiplas vezes no sistema e que, portanto, se repetem na tabela de dados.

Cada entrada na tabela possui valores para cada uma das colunas. Ao realizar o *merge* de dois autores, é possível que ambos possuam valores distintos para diferentes colunas, como Tipo da produção, Ano base, Linha de pesquisa, etc. Para lidar com esses conflitos, são utilizadas as mesmas funções de agregação apresentadas na seção anterior: *Valor mínimo*, *Primeiro valor*, *Lista de valores* e *Contagem de valores*.

O mapeamento das funções de agregação para cada coluna da tabela foi definido da seguinte forma:

- **Tipo da produção:** Primeiro valor
- **Subtipo da produção:** Primeiro valor
- **Ano base:** Primeiro valor
- **Sigla da entidade de ensino:** Contagem de valores
- **Área de concentração:** Contagem de valores
- **Linha de pesquisa:** Contagem de valores
- **Tipo do autor:** Contagem de valores
- **Nome do projeto:** Contagem de valores
- **Id da produção intelectual:** Lista de valores

Sobre as produções intelectuais, é aplicada a técnica de agrupamento por título da produção intelectual, de forma que produções de mesmo título são consideras múltiplas instâncias de uma mesma entidade, sendo agrupadas conforme os critérios definidos acima.

Logo após a importação dos conjuntos de dados, antes da realização do *merge* por títulos, obteve-se um total de 2.505.929 entradas na tabela de produções intelectuais. Após a realização do agrupamento por título, o número de entradas é reduzido para 1.708.666. Isso significa que 31,8% das entradas da tabela possuíam títulos com mais de uma ocorrência no conjunto de dados e foram agrupadas.

É importante lembrar, entretanto, que cada uma das produções intelectuais presentes no *dataset* original possui identificadores (*Id*s) que são referenciados no conjunto de

dados de *Autor da Produção Intelectual de Programas de Pós-Graduação Stricto Sensu no Brasil*. Ao realizar o agrupamento de produções intelectuais, todas as produções agrupadas serão representadas por um único identificador. Os demais identificadores, entretanto, não podem ser descartados, visto que ainda são referenciados no *dataset* de autores e precisam ser atualizados. Por esse motivo, no processamento de dados de autores, é exportado um arquivo adicional no formato *.json* que cumpre o papel de dicionário substituições de Ids de produções, denominado *prod_id_replacements.json*. Este arquivo nada mais é do que um dicionário no formato chave – valor. Para cada identificador que foi substituído por outro durante o processo de agrupamento de produções, existe uma entrada no dicionário de substituições com o seguinte formato:

```
{ "ID_ANTIGO":  "ID_NOVO"}
```

Ao final do processamento dos dados referentes a produções intelectuais, são produzidos dois arquivos de saída:

- **processed_productions.csv:** Contém o conjunto final de produções intelectuais processados nesta etapa.
- **prod_id_replacements.json:** Contém um mapa de substituição de identificadores de produções intelectuais a serem substituídos na tabela de autores.

3.2.3 Conjunto de Dados de Programas da Pós-Graduação

O *dataset* de programas de pós-graduação utilizado no trabalho é composto por um conjunto de 4 arquivos no formato *.csv*, sendo eles:

- programas-2017.csv (4.347 registros, 1,7 MB)
- programas-2018.csv (4.363 registros, 1,7 MB)
- programas-2019.csv (4.570 registros, 1,8 MB)
- programas-2020.csv (4.559 registros, 1,8 MB)

Cada um desses arquivos contém uma lista de entradas correspondendo a programas de pós-graduação brasileiros cadastrados no sistema CAPES referente ao ano em questão. Para cada programa de pós-graduação, existe uma linha no *dataset* especificando 32 propriedades a respeito do programa em questão.

PRÉ-PROCESSAMENTO

O pré-processamento desses dados se inicia pela concatenação dos diversos arquivos em uma tabela única de programas de pós-graduação. Em seguida, as colunas da tabela são filtradas, selecionando apenas as 8 colunas de interesse descritas da seção 3.1.1 dentre as 32 colunas disponíveis no *dataset* original.

AGRUPAMENTO

Após a concatenação dos diversos arquivos que compõem o conjunto de dados de programas de pós-graduação, é possível observar múltiplas ocorrências de um mesmo programa. Isso ocorre pois ao concatenar arquivos de dados referentes a quatro anos de apuração (2017-2020), há instâncias repetidas daqueles programas que existiram por mais de um ano nesse período. Nota-se, entretanto, que os valores campo *Código do programa IES* se mantêm consistentes ao longo dos anos, de forma que um agrupamento com base nessa coluna é suficiente para eliminar a multiplicidade de entradas na tabela.

Entretanto, cada entrada na tabela possui valores possivelmente distintos para cada uma das colunas. Ao realizar o *merge* de dois programas, adotou-se uma metodologia mais simples de seleção de colunas: foi selecionada apenas a última ocorrência de cada programa (último ano no qual ele aparece no conjunto de dados). Essa escolha foi feita por entender-se que a última ocorrência possui o conjunto de informações mais recente (e portanto atualizado) a respeito do programa de pós-graduação em questão.

Ao final do processamento dos dados referentes aos programas de pós graduação, é produzido um arquivo de saída:

- **processed_programs.csv:** Contém o conjunto final de programas de pós-graduação processados nesta etapa.

3.2.4 Conjunto de Dados de Cursos da Pós-Graduação

O *dataset* de cursos de pós-graduação utilizado no trabalho é composto por um conjunto de 4 arquivos no formato *.csv*, sendo eles:

- cursos-2017.csv (6.494 registros, 2,2 MB)
- cursos-2018.csv (6.695 registros, 2,3 MB)
- cursos-2019.csv (6.950 registros, 2,4 MB)
- cursos-2020.csv (7.000 registros, 2,4 MB)

Cada um desses arquivos contém uma lista de entradas correspondendo a cursos

de pós-graduação brasileiros cadastrados no sistema CAPES referente ao ano em questão. Para cada programa de pós-graduação existe uma linha no *dataset* especificando 28 propriedades a respeito do programa em questão.

É possível perceber que cursos de pós-graduação não fazem parte da lista de entidades elencadas no início deste capítulo para compor a rede de colaborações acadêmicas. O motivo do processamento deste *dataset* é o conjunto de colunas com informações a respeito de universidades brasileiras que ele possui. Universidade são entidades que farão parte da rede de colaborações acadêmicas e a CAPES não disponibiliza um conjunto de dados específico com informações a respeito dessas instituições. A solução encontrada foi extrair os dados necessários do *dataset* de cursos universitários.

PRÉ-PROCESSAMENTO

O pré-processamento desses dados se inicia pela concatenação dos diversos arquivos em uma tabela única de cursos de pós-graduação. Em seguida, as colunas da tabela são filtradas, selecionando apenas as 8 colunas de interesse descritas da seção 3.1.1 dentre as 28 colunas disponíveis no *dataset* original.

AGRUPAMENTO

Neste processamento, busca-se apenas informações sobre as universidades brasileiras. O conjunto de dados contém uma entrada para cada curso de pós-graduação pertencente a uma universidade brasileira e cada uma dessas entradas possui informações a respeito da instituição de ensino como *Nome da região*, *Status jurídico*, *Código da entidade CAPES*, etc. Dessa forma, apenas uma entrada por instituição de ensino brasileira é suficiente para obtermos toda a informação necessária. A solução encontrada foi agrupar as entradas da tabela pela coluna *Sigla da entidade de ensino*.

Entretanto, cada entrada na tabela possui valores possivelmente distintos para cada uma das colunas de interesse. Ao realizar o *merge* de cursos de uma mesma instituição, adota-se como metodologia selecionar a última ocorrência de cada curso pertencente a instituição de interesse. Essa escolha foi feita por entender-se que a última ocorrência possui o conjunto de informações mais recente (e portanto atualizado) a respeito do programa de pós-graduação em questão.

Ao final do processamento dos dados referentes aos programas de pós-graduação, é produzido um arquivo de saída:

- **processed_universities.csv:** Contém o conjunto final de entidades de ensino processados nesta etapa.

3.2.5 Pós-Processamento de dados

Esta etapa de pós-processamento dos dados ocorre após a conclusão do processamento individual de cada um dos conjuntos de dados utilizados no trabalho. Isso ocorre pois esta etapa depende de produtos de alguns desses processamentos. As ações desempenhadas nessa etapa são detalhadas a seguir:

1. **Substituição de identificadores de produções intelectuais:** Na seção a respeito do processamento do conjunto de dados de produções intelectuais, foi informado que, ao final do processo, juntamente com o conjunto processado de produções intelectuais, é exportado um arquivo chamado *prod_id_replacements.json* contendo um mapa de substituição de *Id*s de produções intelectuais que, por terem sido agrupados com outras produções, deixaram de existir. Esta etapa do pós-processamento substitui todas as referências a esses identificadores dentro do conjunto de dados de autores, certificando que todas as referências a produções acadêmicas dentro da rede de colaborações sejam válidas.
2. **Extração de informações de linha de pesquisa dos autores:** Uma das informações de interesse a respeito dos autores é a sua *Linha de pesquisa*. Essa informação, entretanto, não faz parte do conjunto de dados de autores de produções intelectuais, estando presente apenas no *dataset* de produções intelectuais. Esta etapa do processamento visa extrair dados de *Linha de pesquisa* dos autores com base nas produções intelectuais por ele autoradas. No caso de haverem divergências entre as linhas de pesquisa que classificam diferentes trabalhos de um determinado autor, escolhe-se aquela que se repete com maior frequência.
3. **Expansão de produções dos autores:** No presente momento, a tabela de autores possui uma única linha para cada indivíduo, contendo uma lista com os identificadores suas produções intelectuais. Essa etapa transforma cada par autor – produção em uma única linha da tabela. Isso é feito para otimizar o processo de importação desses dados no banco de dados de grafos em uma etapa futura.
4. **Geração de mapa de co-autorias:** Também com o propósito de otimizar a importação dos dados para o banco de dados de grafo na próxima etapa do trabalho, realizou-se o processamento de co-autorias entre autores de forma a gerar um arquivo auxiliar no formato *.csv* contendo entradas no formato: `AUTOR_1;AUTOR_2;PROD_ID`. Cada entrada representa uma relação de co-autoria entre dois autores, indicando que

os autores de *Id* AUTOR_1 e AUTOR_2 são ambos autores da produção intelectual de *Id* PROD_ID.

Ao final desta etapa de pós-processamento dos dados, são produzidos dois arquivos de saída:

- **final_authors.csv:** Contém o conjunto final de autores processados nesta etapa.
- **co_authorships.csv:** Contém um mapa auxiliar de co-autorias entre autores.

3.3 Banco de Dados de Grafos

Após o término da etapa de processamento de dados, tem-se como produto arquivos no formato *.csv* contendo os dados processados a respeito de autores, produções intelectuais, programas de pós-graduação e instituições de ensino superior. O objetivo desde trabalho, no entanto, é a consolidação de uma ferramenta para visualização interativa desta rede de colaborações acadêmicas, de forma que necessita-se de meios para a realização de consultas dinâmicas a respeito de subconjuntos de entidades do *dataset* bem como relações entre elas.

Para atingir esse objetivo, decidiu-se utilizar a ferramenta de gerenciamento de banco de dados de grafos Neo4j[3] (NEO4J, 2022a). O Neo4J é o gerenciador de bancos de dados de grafos mais utilizado no mundo, oferecendo suporte a consultas eficientes mesmo para conjuntos de dados de larga escala. O banco de dados é capaz de armazenar dois tipos de entidades: *nós* e *relacionamentos*. Para criar um grafo de colaborações acadêmicas de larga escala a partir dos dados processados na etapa anterior, iremos popular o banco de dados com as seguintes entidades:

3.3.0.1 Nós

- **Autor:** Este nó representa um autor de produções intelectuais na nossa rede e possui as seguintes propriedades:
  - Id: Identificador do autor
  - Id capes: Identificador do autor na base de dados CAPES
  - Nome: Nome completo do autor
  - Nome ABNT: Abreviação do nome do autor no formato ABNT

[3] https://neo4j.com/

  - Programa IES: Programa de pós-graduação do qual o autor faz parte
  - Id programa IES: Identificador do programa do qual o autor faz parte
  - Linha de pesquisa: Linha de pesquisa do autor
  - Tipo: Tipo do autor (docente, discente, etc)
  - Universidade: Sigla da instituição de ensino
  - Contagem de produções: Número de produções de autoria do autor

- **Programa:** Este nó representa um programa de pós-graduação na rede e possui as seguintes propriedades:
  - Id: Identificador do programa
  - Nome: Nome do programa
  - Nome completo: Nome completo do programa
  - Área do conhecimento: Área do conhecimento na qual o programa atua
  - Sub-área do conhecimento: Sub-divisão mais específica da área do conhecimento
  - Grande área do conhecimento: Grande área do conhecimento do programa
  - Contagem de produções: Número de produções produzidas pelo programa
  - Área de avaliação: Área de avaliação do programa de pós-graduação
  - Especialidade: Especialidade do conhecimento programa de pós-graduação
  - Universidade: Sigla da instituição de ensino

- **Universidade:** Este nó representa uma instituição de ensino superior na rede e possui as seguintes propriedades:
  - Id: Identificador da instituição de ensino superior
  - Nome: Sigla da instituição
  - Nome completo: Nome completo da instituição
  - Status jurídico: Classificação do Status Jurídico. (Ex.: federal, estadual, particular, ...)
  - Contagem de produções: Número de produções produzidas pela instituição
  - Região: Nome da região da instituição. (Ex.: sul, sudeste, norte, ...)
  - UF: Sigla da unidade federativa
  - Município: Nome do município

- **Produção:** Este nó representa uma produção intelectual na rede e possui as seguin-

tes propriedades:

- Id: Identificador da produção
- Nome: Título da produção intelectual
- Ano: Ano de publicação
- Tipo: Tipo da produção (Ex.: bibliográfica)
- Sub-tipo: Sub-tipo da produção (Ex.: artigo em periódico)
- Áreas de foco: Áreas de foco da produção, com contagem de ocorrências
- Programas IES: Programas de pós-graduação, com contagem de ocorrências
- Projetos: Projetos de pesquisa, com contagem de ocorrências
- Linhas de pesquisa: Linhas de pesquisa, com contagem de ocorrências
- Universidades: Instituições de ensino, com contagem de ocorrências

3.3.0.2 Relacionamentos

- **Autoria:** Este é um relacionamento entre nós *Autor* -> *Produção*, indicando um autor de uma produção intelectual. Este relacionamento não possui propriedades.
- **Colaboração:** Este é um relacionamento entre nós *Universidade* <-> *Universidade*, indicando que ambas as universidades colaboram entre si e possui a seguinte propriedade:
  - Contagem de colaborações: Número de colaborações entre o par de universidades.
- **Co-autoria:** Este é um relacionamento entre nós *Autor* <-> *Autor*, indicando que os autores compartilham a autoria de uma produção intelectual e possui as seguintes propriedades:
  - Contagem de colaborações: Número de colaborações entre o par de autores.
  - Colaborações: Lista de identificadores das produções acadêmicas em que os autores colaboram.
- **Filiação a programa:** Este é um relacionamento entre nós *Autor* -> *Programa*, indicando que o autor está associado ao programa de pós-graduação. Este relacionamento não possui propriedades.
- **Filiação a universidade:** Este é um relacionamento entre nós *Autor* -> *Universidade*, indicando que o autor está associado à instituição de ensino superior. Este

relacionamento não possui propriedades.

Uma vez decidido o *schema* da base de dados de grafo, com seus nós, relacionamentos e propriedades correspondentes, é preciso importar os dados produzidos na etapa de processamento de dados (seção 3.2).

A importação de dados para o banco é feita utilizando *Cypher*, uma linguagem de consultas inspirada em SQL e projetada para a interação com nós e relacionamentos de grafos. Exemplos de importação de dados podem ser vistos na seção 2.3.2, onde é feita uma análise da ferramenta baseada na criação de um grafo simplificado de co-autorias acadêmicas. Na análise feita, foi possível observar um elevado tempo de importação dos dados, além de um aumento não linear do tempo de importação com o aumento do conjunto de dados. Enquanto durante a realização dos testes foram usados conjuntos de dados com poucas dezenas de milhares de entidades, a rede final que pretende-se construir conta com milhões de entidades. Nesse contexto, é evidente que é preciso otimizar o processo de importação de dados a fim de lidar com o grande volume de dados.

A seguir, são detalhadas uma série de otimizações realizadas sobre o processo de importação de dados ao Neo4J descrito na seção 2.3.2, de forma a tornar o processo eficiente e possibilitar em importação de grandes volumes de dados em tempo hábil. A primeira etapa do esforço de otimização consiste na análise do processo atual de importação com a finalidade de identificar pontos de gargalo, ou seja, os motivos que levam o processo de importação de dados a consumir elevados recursos de tempo computacional.

Após a realizações de alguns testes envolvendo a importação de diferentes volumes de dados, foi possível observar que o tempo de importação aumenta significativamente uma vez que o processo passa a usar uma quantidade maior de memória RAM do que o limite máximo definido para o *Heap* da aplicação, que na configuração padrão é definido como 1 *Gigabyte*. Uma vez que os dados em memória ultrapassam este limite, a aplicação passa a utilizar recursos de paginação em disco, reduzindo o desempenho do processamento. Com o objetivo de otimizar a utilização do recurso de memória RAM durante a importação dos dados forma tomadas duas medidas.

A primeira consiste na utilização do recurso de *periodic commits* do Neo4J. Por padrão, uma *query* Cypher é executada por completo antes que uma transação seja realizada com a finalidade de persistir os resultados da execução. Enquanto para pequenas quantidades de dados este comportamento não representa problemas, a execução de uma *query* dependente de grandes quantidades de dados implica que a totalidade dos dados deve permanecer em memória até a conclusão do processo, com uma única transação de

escrita envolvendo grandes quantidades de dados. O uso do recurso de *periodic commits* permite que seja determinado o número máximo de escritas a serem acumuladas e mantidas em memória antes que uma transação de escrita seja realizada. Isso nos possibilita escrever *queries* de forma que não ultrapassem o limite de memória RAM reservado para o *heap* do processo.

Outra medida tomada foi a alterações das configurações do Neo4J de forma a disponibilizar mais recursos de memória para uso da aplicação. O aumento da memória disponível para o sistema de banco de dados não só otimiza a importação dos dados como também otimiza a operação do banco de dados, permitindo que retorne consultas com maior eficiência e suporte um número maior de requisições simultâneas. Para isso, foi editado arquivo de configurações *neo4j.conf* (por padrão localizado em */etc/neo4j/neo4j.conf*). As seguintes propriedades foram atualizadas:

- **dbms.memory.heap.initial_size**
  - Descrição: Determina o espaço de memória RAM alocado inicialmente para o *heap* do processo.
  - Valor padrão: 512m
  - Valor atualizado: 8G
- **dbms.memory.heap.max_size**
  - Descrição: Determina o espaço máximo de memória RAM a ser alocado para o *heap* do processo.
  - Valor padrão: 1G
  - Valor atualizado: 8G
- **dbms.memory.pagecache.size**
  - Descrição: Determina o espaço máximo de memória RAM utilizado como área de *cache* de dados em disco.
  - Valor padrão: 512m
  - Valor atualizado: 4G

Essas otimizações aceleraram significativamente o processo de importação dos dados. Para fins de comparação, foi selecionada uma *query* responsável pela importação e criação dos nós de autores, criação dos nós de produções intelectuais com base nos identificadores referenciados no *dataset* de autores, bem como criação dos relacionamentos de autoria entre autores e produções. O conjunto de dados utilizado é a tabela de auto-

res resultante da etapa de processamento de dados, contendo 1.275.852 autores que farão parte do grafo de colaborações gerado neste trabalho.

Com a finalidade de comparar o tempo de importação do conjunto de dados com as diferentes técnicas de otimização, executou-se a *query* em 3 cenários: sem otimização (512M *heap size*), com aumento da memória (8G *heap size*) e, por fim, combinando o aumento da memória (8G *heap size*) com o uso da técnica de *commits* periódicos. Ao executar o processo de importação, entretanto, apenas o último cenário teve a sua execução bem sucedida. Os demais cenários, aqueles que não utilizam a técnica de *commits* periódicos, apresentaram erro de falta de memória e não foram concluídos. Isso porque mesmo após o aumento do espaço de memória disponível para a aplicação para 8GB, o processo de importação utiliza uma quantidade maior de memória do que aquela disponível na máquina. Ao analisar o tempo de importação após o aumento da memória disponível juntamente com a habilitação de *commits* periódicos a cada 20.000 produções, obteve-se um tempo de importação bastante satisfatório de 5 minutos e 21 segundos.

Essas otimizações se mostraram efetivas para o processo de importação de todos os tipos de nós definidos no *schema* da base de dados e para a importação da maioria dos relacionamentos entre nós. A *query* relativa à geração de um tipo de relacionamento *co-autorias*, entretanto, apresentou um tempo de processamento bastante ineficiente. Enquanto todas as demais *queries* foram processadas dentro de alguns minutos, a geração das relações de co-autoria entre autores não foi concluída dentro de dias de processamento quando aplicada ao conjunto de dados completo de autores. Isso porque, em um primeiro momento, as informações de co-autorias não eram exportadas durante a etapa de processamento de dados, sendo geradas em tempo de importação a partir de consultas complexas na linguagem *Cypher* envolvendo grande quantidades de buscas no banco de dados e usa intenso de recursos de memória.

A solução adotada foi o cálculo das informações de co-autoria durante a etapa de pós-processamento dos dados, como descrito na seção 3.2.5. Dessa forma, é exportado o arquivo *co_authorships.csv* contendo um mapa de todas as co-autorias entre autores, de forma que a importação pode ser feita de forma eficiente (concluída em 22 minutos e 4 segundos) através da *query* a seguir:

```
LOAD CSV WITH HEADERS FROM 'file:///co_authorships.csv'AS row
FIELDTERMINATOR ';'
CALL {
```

```
  WITH row
  WITH
    row.AUTHOR_1 as a1_id,
    row.AUTHOR_2 as a2_id,
    row.PROD_ID as prod_id
  MATCH
    (a1:Author {id: toInteger(a1_id)}),
    (a2:Author {id: toInteger(a2_id)})
  MERGE (a1)-[coauthor:CO_AUTHOR]-(a2)
  ON CREATE SET
    coauthor.collabs_count = 1,
    coauthor.collaborations = [prod_id]
  ON MATCH SET
    coauthor.collabs_count = coauthor.collabs_count + 1,
    coauthor.collaborations = coauthor.collaborations + [prod_id]
} IN TRANSACTIONS OF 20000 ROWS;
```

O conjunto completo de *queries* que compõem o processo de importação dos dados estão disponíveis no repositório *GitHub*[4] do trabalho. Após as otimizações, o processo completo de importação dos dados para a base de dados Neo4j foi concluído em 33 minutos e 36 segundos, produzindo uma rede de colaborações acadêmicas composta por:

- 1.275.852 autores
- 1.708.666 produções intelectuais
- 4.685 programas de pós-graduação
- 532 instituições de ensino superior
- 6.072.199 relações de autoria ([autor] -> [produção])
- 14.883.507 relações de co-autoria entre autores
- 1.140.188 relações de colaboração entre instituições de ensino
- 1.275.852 relações de filiação a programas de pós-graduação ([autor] -> [programa])
- 1.275.852 relações de filiação a instituições de ensino ([autor] -> [universidade])

[4]O código fonte completo deste trabalho está disponível em https://github.com/ComputerNetworks-UFRGS/vizcolab

4 APLICAÇÃO DE VISUALIZAÇÃO DA REDE

Esta etapa do trabalho desenvolve uma aplicação web para a exploração dinâmica da rede de colaborações gerada na etapa anterior. O software resultante, denominado VizColab, permite a exploração de relações de co-autoria de trabalhos acadêmicos em um grafo tri-dimensional com representações de colaborações entre entidades em 3 níveis distintos: universidades, programas de pós-graduação e autores. Em cada um dos níveis de visualização, o usuário é capaz de explorar um grafo onde entidades (universidades, programas ou autores) são modelados como nós, enquanto arestas entre os nós representam colaborações entre as duas entidades. O usuário é capaz de selecionar qualquer entidade para visualizar detalhes a seu respeito, arrastar nós de forma a reposicioná-los no grafo, mover a câmera com movimentos de rotação, zoom ou deslocamento. É apresentada ainda a possibilidade de explorar um determinado nó, revelando um novo grafo em um outro nível de representação, contendo entidades pertencentes ao nó selecionado, bem como entidades com quem elas colaboram.

Figura 4.1 – Aplicação VizColab - Visualização dos autores do programa de computação da UFRGS

4.1 Arquitetura da aplicação

O software de visualização da rede desenvolvido foi nomeado VizColab[1] e consiste em uma aplicação *web* do tipo *Single Page Application* (MDN, 2021), construída usando o *framework* reativo *React* (META, 2022). A aplicação faz uso da biblioteca de visualização de grafos *3D Forces Graph* (ASTURIANO, 2017) para a exibição de grafos tri-dimensionais de colaborações acadêmicas em três níveis hierárquicos distintos: universidades, programas de pós-graduação e autores de produções intelectuais.

Por se tratar de uma *Single Page Application*, a aplicação web pode ser servida como um simples conjunto de arquivos estáticos e tem a capacidade de se conectar diretamente à base de dados de grafos Neo4j para a obtenção de dados sob-demanda através do conector *JavaScript* disponibilizado pela ferramenta, não havendo necessidade de um servidor *back-end* para intermediar as requisições. A arquitetura de alto nível da aplicação contendo os componentes necessários para o seu funcionamento pode ser visualizada do diagrama da Figura 4.2.

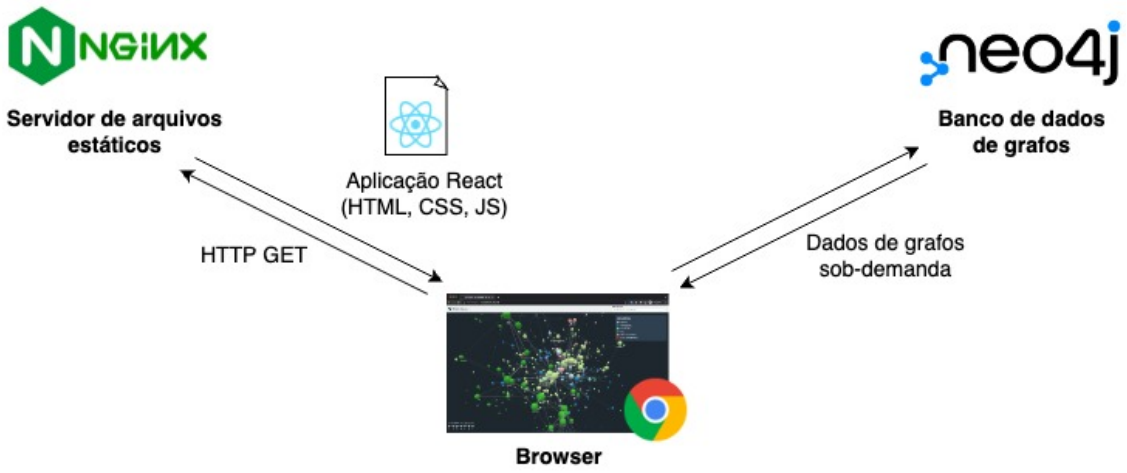


Figura 4.2 – Arquitetura de alto nível da aplicação VizColab

4.2 Técnicas para a visualização do grafo de larga escala

Enquanto a implementação de uma visualização para grafos pequenos ou médios pode ser alcançada facilmente através do carregamento de seus dados em uma biblioteca gráfica, visualizar grafos de larga escala não é uma tarefa trivial. Isso porque o número de arestas em grafos altamente conectados cresce de forma exponencial com o aumento do

[1] A ferramenta está disponível em http://vizcolab.inf.ufrgs.br/

número de nós (como ilustrado na Figura 4.3), causando dois problemas principais. O primeiro deles diz respeito ao armazenamento em memória dos dados que compõem o grafo e das representações gráficas de seus componentes, que para grandes grafos altamente conectados pode facilmente se tornar maior do que a memória disponível nos dispositivos de hardware em uso atualmente. O segundo diz respeito ao elevado número de conexões entre nós, que se aglomeram e sobrepõem aos demais elementos quando renderizados para grandes grafos, impedindo a compreensão.

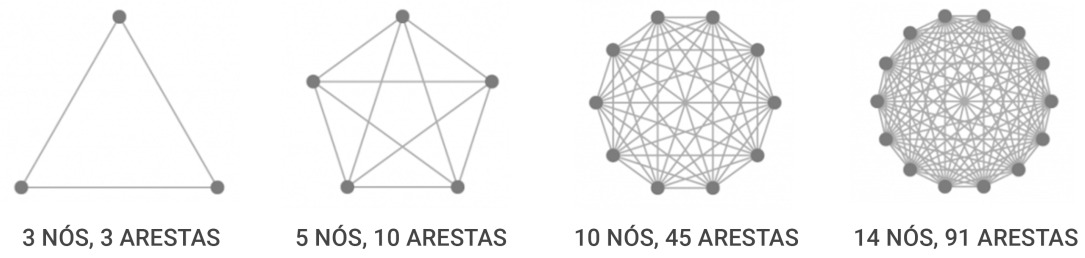


Figura 4.3 – Ilustração do crescimento exponencial do número de conexões em decorrência do aumento linear do número de nós em um grafo completo

A aplicação desenvolvida contorna esses problemas através da implementação de duas técnicas principais: segmentação hierárquica de nós e densidade de conexões variável. Ambas as técnicas serão detalhadas a seguir.

4.2.1 Segmentação hierárquica de nós

A rede de colaborações acadêmicas proposta é composta por milhões de nós e dezenas de milhões de arestas. Nesse cenário, é fácil perceber que a abordagem trivial de renderização da rede completa para os usuários não é viável, tanto pela perspectiva computacional quanto pela perspectiva visual. Para viabilizar a visualização dessa rede, algumas alternativas foram consideradas.

A primeira delas foi a utilização de uma técnica de ocultação de nós distantes do observador, de forma a carregar novos nós dinamicamente sempre que a câmera for movimentada pelo observador. Embora a técnica possa parecer promissora em uma primeira análise, algumas de suas características prejudicam a experiência do usuário e acrescentam complexidade à implementação da solução. Em primeiro lugar, a requisição de nós e arestas com base na sua posição requer que todas as posições de nós e arestas no espaço gráfico tri-dimensional sejam pré-calculadas e armazenadas no banco de dados de grafos.

Essa abordagem, entretanto, impede que o usuário possa movimentar nós durante sua interação com o grafo, visto que uma reconfiguração em tempo real da posição dos nós e arestas tornaria as posições pré-calculadas dos demais nós no banco de dados obsoletas, prejudicando a experiência interativa da aplicação. O segundo problema que surge ao ocultar nós distantes do observador diz respeito à forma de lidar com as arestas entre nós visíveis e nós ocultos. Optar por manter essas arestas visíveis no grafo implica em uma grande quantidade de conexões entre nós visíveis e potencialmente centenas de milhares de outros nós ocultos, obstruindo a visualização e necessitando que informações a respeito da posição de todos os nós visíveis e invisíveis envolvidos nessas conexões sejam mantidas na memória da aplicação cliente. Por outro lado, ao optar-se por ocultar arestas envolvendo nós ocultos, a visualização de rede dos nós visíveis torna-se pouco confiável e significativa, uma vez que boa parte de suas co-autorias podem estar invisíveis pelo simples fato desses co-autores habitarem uma posição distante no espaço tri-dimensional pré-calculado do grafo. Por fim, a existência de um grafo único de autores dispersos por uma ampla região tri-dimensional torna a tarefa de identificar autores pertencentes a uma instituição de ensino ou a um determinado programa de pós graduação praticamente inviável em meio a inúmeros nós e arestas dispersos através do grande grafo.

Por conta dessas características, a alternativa de implementar a técnica de ocultação de nós distantes foi descartada em detrimento do uso de uma nova técnica: a segmentação hierárquica de nós. Este método consiste na adição de dois novos níveis hierárquicos de visualização de colaborações à nossa rede de co-autorias. Além da visualização de colaborações acadêmicas entre autores, adicionamos à aplicação visualizações de colaborações entre instituições de ensino superior e entre programas de pós-graduação.

A ideia consiste em mostrar aos usuários, em um primeiro momento, um grafo de colaborações entre instituições de ensino superior (figura 4.4). Ao selecionar um nó referente a essas instituições, o usuário tem a opção de explorá-lo. Essa ação leva o usuário a uma visualização do próximo nível hierárquico, onde é mostrado um grafo de colaborações entre todos os programas de pós graduação pertencentes à instituição de ensino selecionada (figura 4.5). Nós no nível de programas de pós-graduação também podem ser explorados, ação essa que leva o usuário a uma visualização de co-autorias de todos os autores de produções intelectuais filiados àquele programa de pós-graduação, bem como todos os demais autores brasileiros com quem eles colaboram (figura 4.6). Por fim, há ainda a possibilidade de selecionar um autor do grafo de co-autorias e clicar na opção *Explorar*. Essa ação revela o último nível de exibição, que consiste em um grafo contendo

apenas o autor selecionado, seus co-autores e as relações de co-autoria entre esses nós (figura 4.7). Embora essa última visualização não represente um novo nível hierárquico de visualização, visto que ainda consiste em uma visualização de colaborações entre autores, ela foi adicionada com a finalidade de permitir a exploração detalhada da rede de colaborações de um determinado autor.

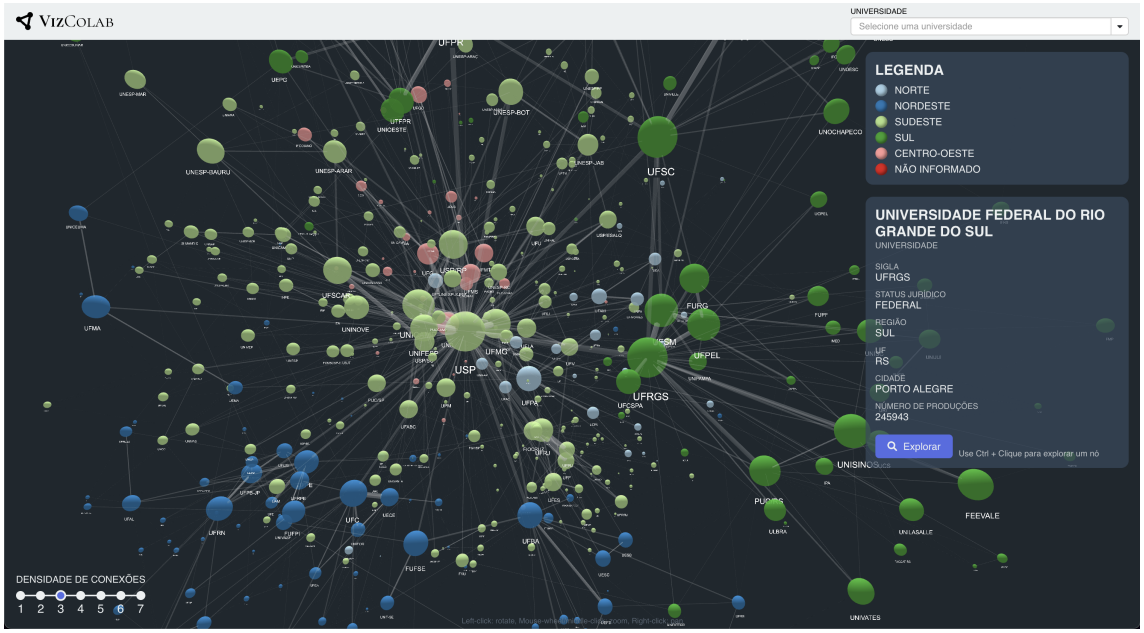


Figura 4.4 – Aplicação VizColab — Visualização ao nível de colaborações entre instituições de ensino superior brasileiras

Enquanto existem mais de um milhão de autores na base de dados, a rede conta com apenas 532 instituições de ensino superior, de forma que todas as instituições podem ser renderizadas simultaneamente em um grafo coeso e navegável. O mesmo vale para a exibição de programas de pós-graduação de uma instituição de ensino ou para autores de um programa de pós-graduação, mantendo o número de nós renderizados dentro de uma faixa segura a fim de garantir a expressividade da visualização do grafo e um bom desempenho de operação. Além disso, o método de segmentação hierárquica ainda enriquece a aplicação ao possibilitar a visualização de colaborações entre instituições de ensino e programas de pós-graduação, aumentando o escopo de uso do software desenvolvido, uma vez que essas novas funcionalidades habilitam a extração de *insights* que não poderiam ser facilmente obtidos com base em um grafo único de colaborações entre autores.

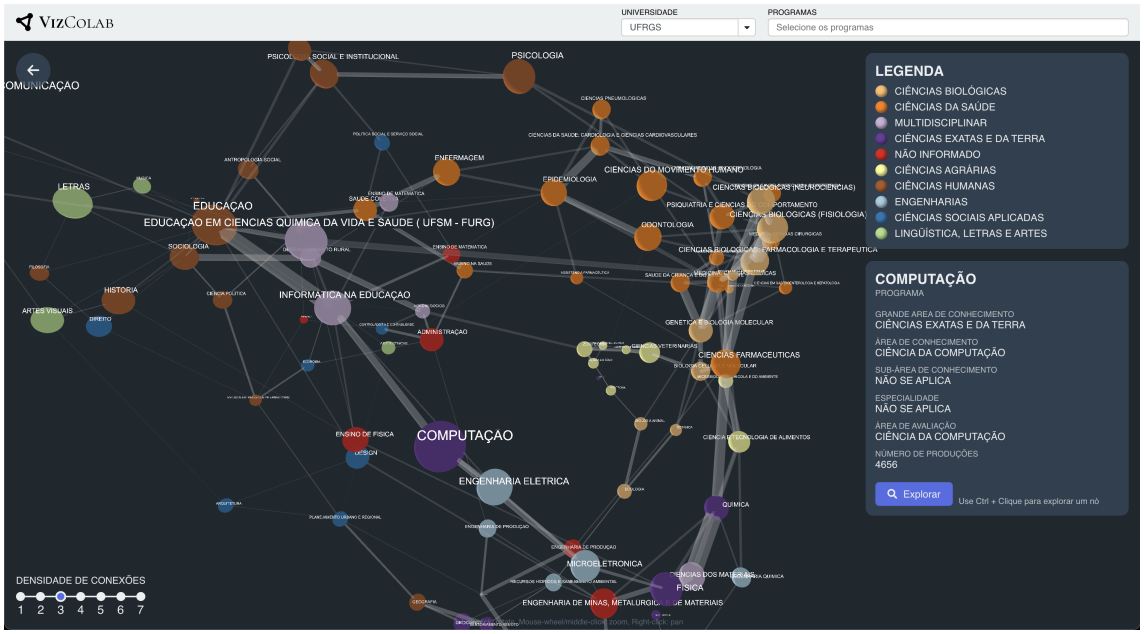


Figura 4.5 – Aplicação VizColab — Visualização ao nível de colaborações entre programas de pós graduação da UFRGS

4.2.2 Densidade de conexões variável

Mesmo com o uso da técnica de segmentação hierárquica de nós do grafo, que reduz o número de nós visualizados simultaneamente, o elevado número de arestas existentes em grafos altamente conectados ainda representa um problema para a visualização de certas redes. Tomando o grafo de colaborações entre instituições de ensino como exemplo, pode-se observar que cada nó do grafo tem colaborações acadêmicas com quase todos os demais. Isso ocorre porque basta que qualquer autor pertencente a uma instituição de ensino tenha colaborado com qualquer outro autor da outra instituição para que uma aresta seja criada entre os dois nós. Embora existam apenas 532 instituições de ensino na base de dados, estas instituições possuem 1.140.188 de arestas de colaboração entre si, de forma que uma eventual renderização de todas essas arestas resultaria em um grafo altamente condensado, impossibilitando a distinção visual de seus elementos.

A fim de contornar este problema, é reciso um método capaz de selecionar as conexões mais relevantes de um grafo dado um parâmetro de densidade. Ou seja, um método que possibilite a variação da quantidade de arestas visíveis um grafo. Para valores baixos de densidade quer-se que sejam visíveis apenas as arestas mais relevantes do grafo. Arestas menos relevantes passam a se tornar visíveis conforme o valor do parâmetro densidade aumenta.

Para satisfazer esses requisitos, foi elaborado o conceito de *densidade de cone-*

Figura 4.6 – Aplicação VizColab — Visualização ao nível de colaborações entre autores ligados ao programa de computação da UFRGS

xões. O método ordena as arestas de cada um dos nós do grafo em ordem decrescente de número de colaborações, de forma que a primeira aresta da lista ordenada será aquela que conecta ao autor com quem se tem o maior número de co-autorias em trabalhos acadêmicos. Em seguida, uma variável de densidade d é definida. Dado qualquer valor inteiro positivo de d, o método retornará uma rede contendo as primeiras d arestas de cada nó do grafo. Dado $d = 1$, por exemplo, o método de densidade retornará um grafo contendo apenas a aresta mais significativa de cada nó do grafo. Para $d = 3$, o grafo resultante contará com as 3 arestas mais significativas de cada um de seus nós e assim por diante. A figura 4.8 compara a visualização do grafo de colaborações entre instituições de ensino antes e depois da técnica de densidade variável de conexões.

A ocultação de nós menos significativos através da utilização do conceito de *densidade de conexões* definido acima possui um conjunto de propriedades favorável ao seu uso no contexto de visualização da rede de colaborações acadêmicas desenvolvida neste trabalho, sendo elas:

- Para qualquer valor de $d >= 1$, sempre existe ao menos uma aresta conectando cada um dos nós do grafo.
- Nós de maior relevância tem mais conexões do que nós de menor relevância, sendo possível identificar *hubs* centrais ou locais no grafo.
- As principais conexões de cada nó do grafo são mantidas (de acordo com o valor

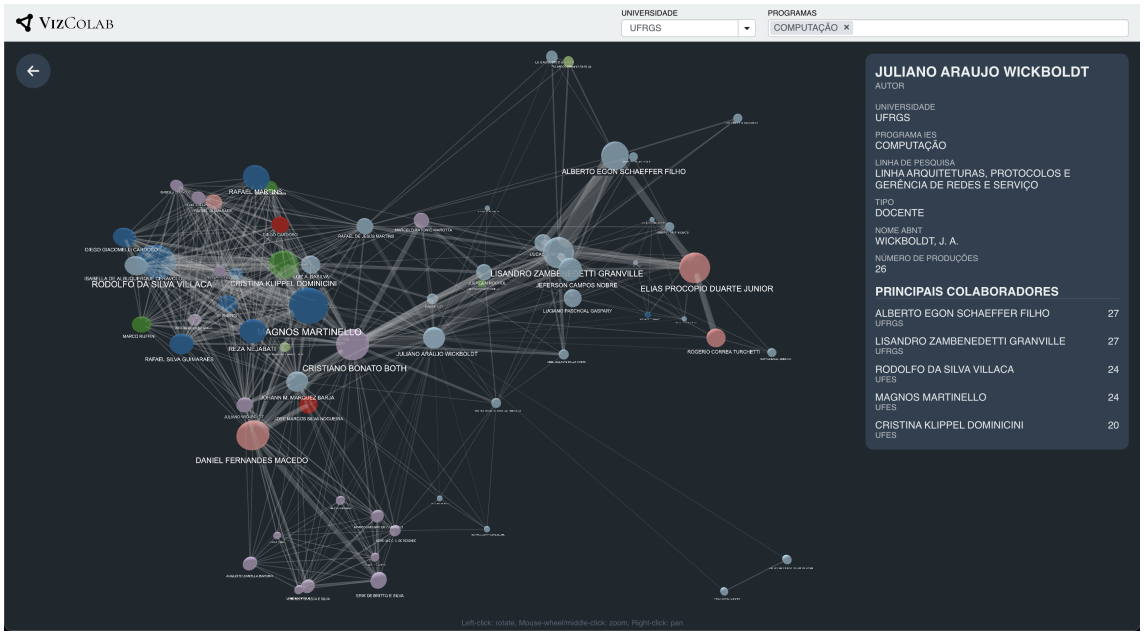


Figura 4.7 – Aplicação VizColab — Visualização de co-autorias entre autores que colaboram com o docente Juliano Araújo Wickboldt

de densidade escolhido) de forma a preservar arestas que apesar de não possuírem grande relevância global, são relevantes localmente entre os nós da qual participam.

Este método foi aplicado a todos os níveis de visualização disponíveis da aplicação (com exceção da visualização de co-autorias de um autor específico, onde são sempre exibidas todas as arestas). A densidade das conexões exibida é definida para o valor 3 por padrão, mas pode ser alterada pelo usuário através de uma barra seletora localizada no canto inferior esquerdo da tela.

4.3 Elementos visuais do grafo

Um grafo tri-dimensional é composto de objetos modelados no espaço a fim de representar linhas e arestas. Esses objetos possuem propriedades visuais capazes de transmitir informações. Para os nós do grafo, pode-se codificar informações em sua forma, tamanho, posição, cor, etc. Para as arestas, é possível codificar informações em sua espessura, comprimento, cor, etc. Esta seção trata de expor as decisões tomadas a respeito dessas características visuais do grafo para que a visualização resultante seja capaz de transmitir a maior quantidade de informações de valor aos usuários, possibilitando o uso da aplicação para a extração de *insights* a respeito de aspectos da colaboração acadêmica nacional.

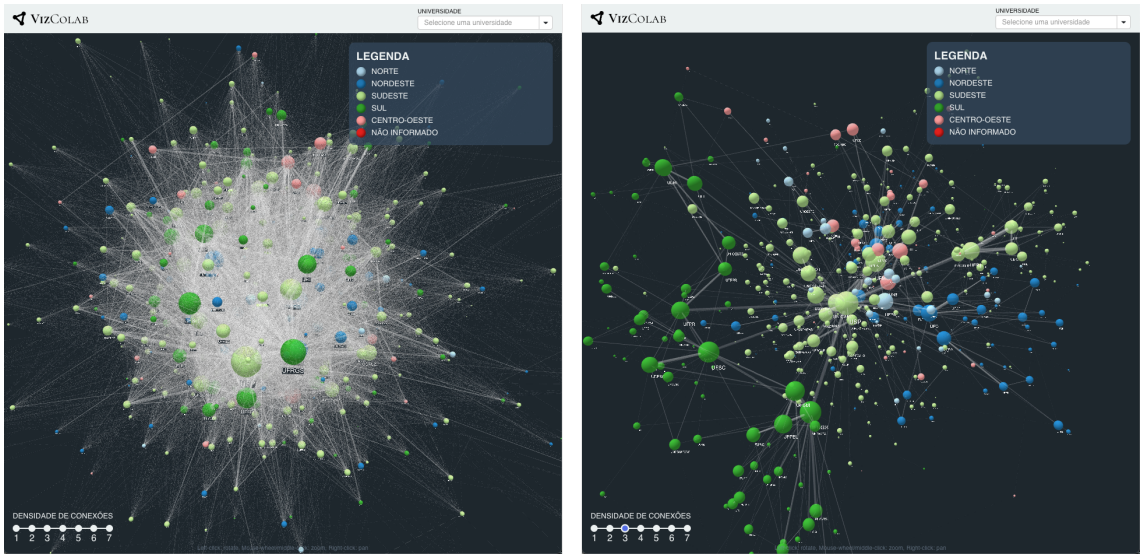


Figura 4.8 – À direita, o grafo de colaborações entre instituições de ensino contendo todas as arestas; À esquerda, o mesmo grafo com o uso de um parâmetro de densidade de conexões de valor 3.

Os nós são representados por esferas no espaço tri-dimensional do grafo. O volume dessas esferas devem traduzir, de certa a forma, a relevância de um nó dentro do grafo, de forma que esferas de maior volume representem nós de maior relevância. Para isso, o volume da esfera foi mapeado para a propriedade *número de produções acadêmicas*, de forma que o volume de cada nó é diretamente proporcional ao número de trabalhos acadêmicos produzidos pela entidade (instituições de ensino, programas ou autores). Outras propriedades como número de citações em trabalhos acadêmicos ou *h-index* foram consideradas para este papel. Essas propriedades entretanto, são dinâmicas e não fazem parte do conjunto de dados da CAPES utilizado neste trabalho. Testes com essas diferentes propriedades podem ser eventualmente realizados em trabalhos futuros.

Outra característica importante dos nós é a sua cor. Para cada nível de visualização, uma propriedade distinta foi escolhida para ser representada pela cor. Para a visualização de instituições de ensino superior, a cor corresponde à região onde a instituição de ensino está localizada (norte, nordeste, sudeste, sul ou centro-oeste). A região foi escolhida por permitir a extração de *insights* a respeito de colaborações entre instituições de diferentes regiões do país. Já para nós de programas de pós-graduação, a propriedade escolhida para ser representada pela cor é a grande área do conhecimento atribuída ao programa em questão (ciências biológicas, engenharias, ciências de saúde, etc). Esta propriedade foi escolhida pela sua abrangência, possibilitando que diversos programas compartilhem o mesmo valor de grande área do conhecimento e possibilitando a formação de *clusters* entre programas altamente conectados. Por fim, para os autores foi escolhida

a propriedade de linha de pesquisa. Esse informação foi escolhida para que seja possível visualizar as tendências de colaboração entre autores de diferentes linhas de pesquisa dentro e fora de um programa de pós-graduação.

As arestas do grafo, por outro lado, são diferenciadas entre si por uma única característica: sua espessura. A espessura de uma aresta é diretamente proporcional ao número de colaborações acadêmicas existentes entre as duas entidades conectadas por ela. No caso de autores, a espessura representa o número de produções intelectuais na qual ambos os nós em suas extremidades são listados como autores, de forma que as linhas mais espessas representam os maiores graus de colaboração entre dois indivíduos. O mesmo vale para colaborações entre programas de pós-graduação e instituições de ensino, onde a espessura de uma aresta é proporcional ao número de trabalhos acadêmicos com indivíduos filiados a ambos os nós em suas extremidades pertencentes à lista de autores.

Por fim, a posição dos nós no espaço também é semanticamente relevante. Isso porque a posição de cada elemento é calculado dinamicamente a partir de um algoritmo de forças. A posição de cada nó no grafo é calculada de acordo com a resultante de um conjunto de três forças que agem sobre ele, sendo elas:

- **Força de conexão:** Esta é uma força de atração existente entre dois nós conectados por uma aresta. A força atrai ambos os nós de forma análoga a força exercida por uma mola entre duas esferas. A intensidade dessa força é proporcional ao número de colaborações representadas pela aresta em questão, de forma que arestas mais espessas exercem maior força de atração do que arestas mais estreitas. Semanticamente, esta força tende a fazer com que nós com maior grau de colaboração entre si sejam dispostos em proximidade da visualização final do grafo.
- **Força de carga:** Esta é uma força exercida por todos os nós do grafo sobre todos os outros. Ela é configurada como uma força de repulsão, funcionando de forma análoga à força eletrostática entre cargas. Quanto maior o volume de um nó (número de produções acadêmicas), maior a força de repulsão exercida por este nó sobre os demais. A existência dessa força é necessária para evitar que todos os nós se aglomerem em um pequeno espaço, visto que todas as demais forças em ação são de atração apenas. Por ser proporcional ao volume do nó, a força de carga também causa um distanciamento entre nós de grande influência na rede, permitindo que cada um mantenha sua própria região de influência ao redor de sua posição.
- **Força central:** Por fim, a força central atrai todos os nós em direção ao ponto central do grafo. Isso evita que grupos de nós se dispersem pelo espaço virtualmente

infinito no qual o grafo está localizado, permitindo que a rede se desenvolva ao redor de um ponto central.

4.4 Exploração da rede de colaborações

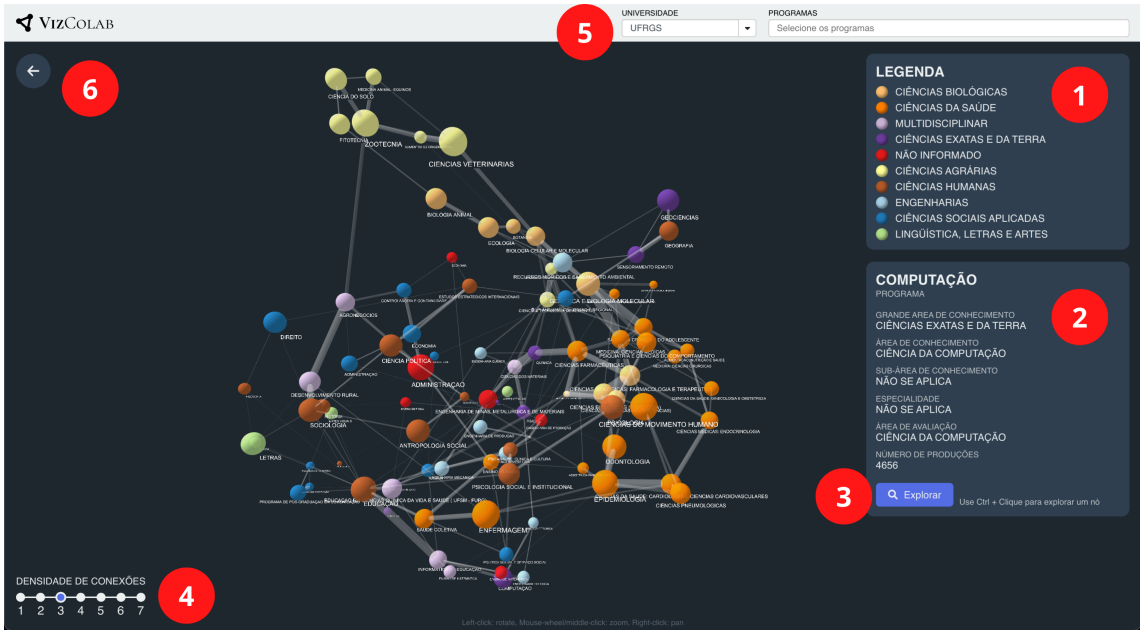


Figura 4.9 – Recursos de exploração da ferramenta VizColab

Além dos grafos de colaborações acadêmicas, a ferramenta VizColab também conta com uma série de recursos projetados para auxiliar o usuário com a navegação, compreensão e exploração da rede de colaborações. A Figura 4.9 enumera seis recursos que compõem a interface gráfica da aplicação:

1. **Painel de legenda:** Este painel se encontra à direita da tela e apresenta uma legenda de cores para cada um dos diferentes valores encontrados no grafo em exibição.
2. **Painel de informações:** Este painel, localizado abaixo do painel de legenda, exibe informações detalhadas a respeito da entidade selecionada no grafo (uma entidade é selecionada ao ser clicada com o botão esquerdo do mouse). As informações exibidas variam de acordo com o nível da visualização (entidades de ensino, programas ou autores).
3. **Botão explorar:** Ao final do painel de informações, está disponível o botão explorar. Ao clicar este botão, o usuário passa a visualizar um novo nível de grafo contendo entidades que fazem parte do nó selecionado. Em uma visualização de instituições de ensino, por exemplo, ao selecionar a universidade UFRGS no grafo

e clicar no botão explorar, o usuário passará a visualizar um novo grafo contendo colaborações entre os programas de pós-graduação que compõem a universidade selecionada. Ao selecionar um programa de pesquisa e clicar no botão explorar, o usuário passará a visualizar um grafo de colaborações entre os autores que fazem parte do programa de pesquisa selecionado, incluindo autores de outros programas ou universidades com quem eles colaboram. Por fim, ao selecionar um autor e clicar no botão explorar, o usuário passa a visualizar um grafo contendo apenas o autor selecionado e os demais autores com quem ele colabora, permitindo a exploração detalhada de suas relações de co-autoria.

4. **Seletor de densidade de conexões:** Localizado no canto inferior esquerdo da tela, este seletor permite a variação do parâmetro *densidade de conexões*, descrito em mais detalhes na seção 4.2. A alteração do parâmetro resulta em um grafo com maior ou menor densidade de conexões entre os nós, permitindo ao usuário decidir a visualização mais conveniente ao seu objetivo.
5. **Barra de filtros:** No canto superior direito da tela, estão localizados os campos de filtros. Eles permitem ao usuário pesquisar e selecionar, com ajuda de recursos de sugestão, a instituição de ensino, programa de pós-graduação ou autor que deseja visualizar. O componente ainda permite a seleção de múltiplos programas de pós-graduação simultaneamente, de forma que o usuário pode explorar relações de colaborações entre autores dos diferentes programas.
6. **Botão voltar:** Por fim, no canto superior esquerdo da área do grafo, está localizado o botão *voltar*, que permite ao usuário retornar a visualização para o nível anterior. Isso significa que se o botão for pressionado em uma visualização de colaborações entre autores de um programa, o usuário passará a visualizar o grafo de colaborações entre programas da universidade selecionada. Ao pressionar o botão novamente, o usuário retorná à tela inicial da aplicação, visualizando relações de colaborações entre as instituições de ensino superior brasileiras.

5 ANÁLISE DA SOLUÇÃO

Além de uma ferramenta para a visualização de grafos de co-autoria, a aplicação VizColab tem potencial para ser utilizada como uma importante ferramenta de apoio para indivíduos interessados em compreender aspectos do estado da produção intelectual no Brasil. Sejam esses indivíduos pesquisadores, tomadores de decisão vinculados a poderes públicos ou privados, jornalistas ou até mesmo entusiastas, a ferramenta desenvolvida neste trabalho pode ajudá-los a compreender aspectos como:

- Identificação de núcleos regionais de pesquisa
- Relevância nacional ou local de instituições de ensino
- Focos de pesquisa de instituições de ensino
- Identificação de autores centrais dentro de programas de pós-graduação

Uma vez construída a rede de colaborações e a aplicação de visualização VizColab, inicia-se uma inspeção da rede através da aplicação desenvolvida. O objetivo dessa inspeção é explorar a rede e extrair possíveis *insights* a partir dos grafos de colaboração de acordo com os aspectos elencados acima.

5.1 Identificação de núcleos regionais de pesquisa

O primeiro nível de visualização de colaborações encontrado pelo usuário ao abrir a aplicação VizColab diz respeito a colaborações entre instituições de ensino superior brasileiras (figura 5.1). Com a finalidade de compreender as dinâmicas de interação entre universidades de diferentes regiões do país, os nós do grafo são coloridos de acordo com a região de cada uma das instituições representadas. Em uma primeira análise, pode-se perceber a existência de um agrupamento de nós de mesma cor em regiões distintas do grafo. De acordo com o modelo de forças aplicado sobre as entidades do grafo (detalhado na seção 4.3), a proximidade espacial de um conjunto de nós indica um elevado nível de colaborações acadêmicas entre essas instituições. Dessa forma, pode-se assumir que a aglomeração segmentada de entidades de ensino de uma mesma região em diferentes partes do grafo indica uma alta correlação entre proximidade física entre instituições de ensino e a probabilidade de autores pertencentes a essas instituições colaborarem entre si na concepção de produções intelectuais. Essa suposição é ainda corroborada pelo elevado número de arestas conectando essas instituições, indicando um alto grau de colaborações

entre instituições de uma mesma região do país.

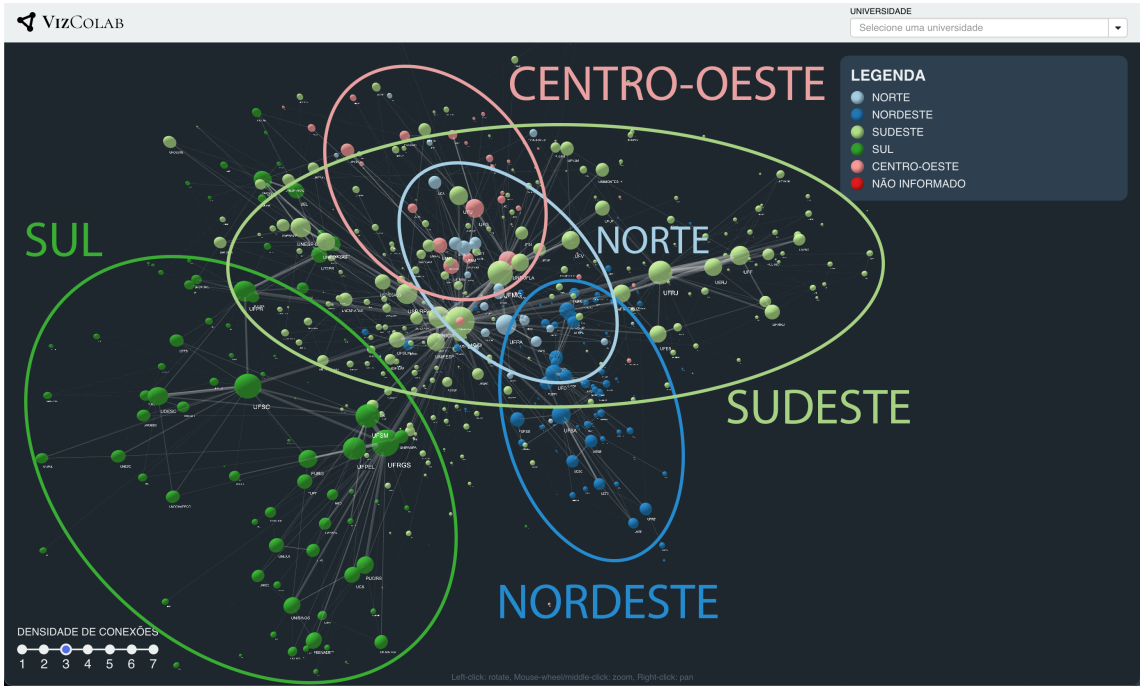


Figura 5.1 – Visualização de instituições de ensino superior no software VizColab, destacando os núcleos regionais.

5.2 Relevância nacional ou local de instituições de ensino

Ainda no nível de visualização de colaborações entre instituições de ensino, ilustrado na Figura 5.1, tentou-se distinguir entre instituições de diferentes relevâncias dentro do cenário da pesquisa nacional ou regional. Ao inspecionar os nós que compõem esse grafo, é possível perceber que uma instituição de ensino se encontra em posição de destaque dentre o conjunto de universidades brasileiras, ocupando um ponto central no grafo: a Universidade de São Paulo (USP). Sua posição de centralidade, entretanto, não é o único aspecto que indica a sua relevância no cenário da pesquisa nacional. É possível observar que a universidade é também representada pela esfera de maior volume no grafo, indicando que a instituição produziu o maior número de trabalhos acadêmicos. Por fim, notamos que esta universidade possui o maior número de conexões com outros nós do grafo, o que segundo o modelo de densidade de conexões (detalhado na seção 4.2) significa que o nó está entre as colaborações mais importantes de muitas outras instituições. A partir da análise desses aspectos do modelo de visualização, conclui-se que a USP se comporta como um *hub* de pesquisa de alta relevância no país.

Ao analisar sub-regiões do grafo, que como mencionado na seção 5.1 são forma-

das por núcleos regionais de instituições, percebe-se que esse padrão se repete diversas vezes em escala menor. Olhando para o núcleo de universidades da região sul do país, por exemplo, identifica-se que a UFRGS se comporta como um desses *hubs* locais. As suas características são bastante semelhantes àquelas elencadas para o *hub* nacional, incluindo volume do nó maior que os demais na região, posição central dentro do sub-grafo regional e elevado número de conexões com outras universidades da região. Além da UFRGS, outras instituições podem ser identificadas como de elevada relevância dentro de suas regiões, como é o caso da UFRJ e UFMG para a região sudeste, UNB e UFG no centro-oeste, UFPE e UFRN no nordeste e a UFPA na região norte do país.

5.3 Focos de pesquisa de instituições de ensino

No que diz respeito às áreas do conhecimento que compõem os focos de pesquisa das instituições de ensino superior, pode-se encontrar contextos bastante variados. Por um lado, existem instituições de ensino generalistas, a exemplo dos *hubs* mencionados na seção anterior, possuindo atividades de pesquisa relevantes em uma ampla gama de áreas do conhecimento. Por outro lado existem também instituições especialistas, que concentram sua pesquisa em um conjunto restrito de áreas do conhecimento, muitas vezes de grande sinergia entre si.

Os recursos de visualização da ferramenta VizColab possibilitam a identificação de instituições generalistas e especialistas. No caso de instituições especialistas, ainda é possível identificar as áreas de foco em que a instituição atua. Como exemplo, pode-se observar o caso da USP. A universidade possui 178 programas de pós-graduação distribuídos entre oito grandes áreas do conhecimento (Figura 5.2). Com base nessa observação, assume-se com certo grau de certeza que esta se trata de uma instituição generalista.

Ao observar, entretanto, instituições como a Universidade Federal de Ciências da Saúde de Porto Alegre (UFCSPA) ou o Instituto Tecnológico de Aeronáutica (ITA) percebemos uma estrutura diferente. Ao contrário da rede de programas que encontramos na USP, essas universidades possuem um número reduzido de programas de pós-graduação altamente relacionados entre si. A UFCSPA possui apenas 11 programas de pós-graduação distribuídos entre três grandes áreas do conhecimento: ciências da saúde, ciências biológicas e ciências humanas. Essas informações, juntamente com uma análise rápida dos 11 programas de pós-graduação oferecidos pela instituição (Figura 5.2), permite assumir que esta é uma instituição especialista com foco em ciências da saúde. O

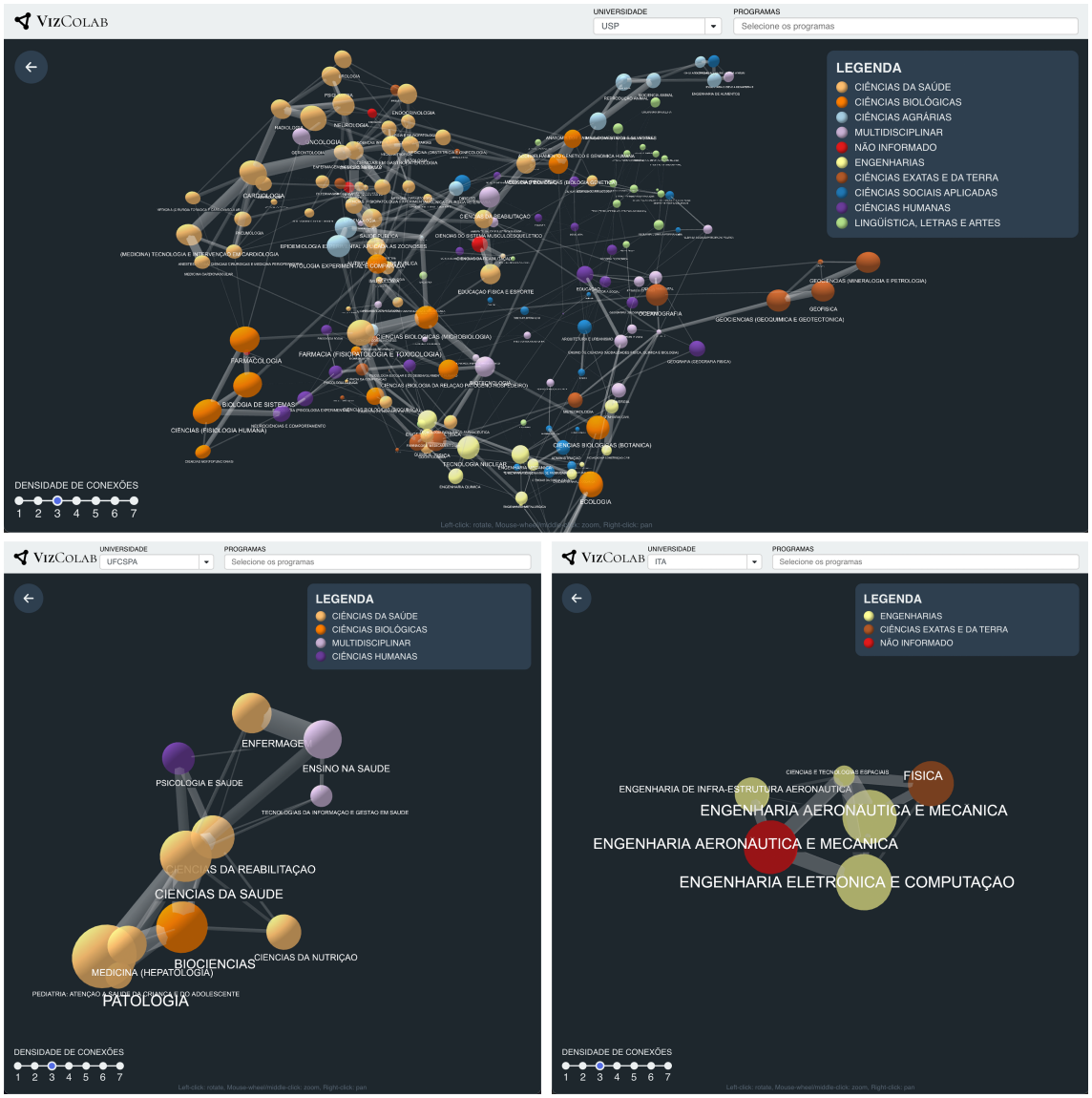


Figura 5.2 – Acima, visualização dos programas de pós-graduação da USP, uma instituição generalista; Abaixo, visualização dos programas de pós-graduação da UFCSPA e ITA, instituições especialistas.

ITA possui uma gama ainda mais restrita com sete programas de pós-graduação distribuídos entre duas grandes áreas do conhecimento: engenharia e ciências exatas e da terra. Assim como para a UFCSPA, essas informações juntamente com uma análise rápida dos 7 programas de pós-graduação oferecidos pela instituição permite assumir que esta é uma instituição especialista com foco em engenharia aeronáutica.

5.4 Identificação de autores centrais dentro de programas de pós graduação

As seções anteriores se concentram na extração de *insights* com base na exploração das visualizações de instituições de ensino e programas de pós-graduação da apli-

cação. Esta seção concentra-se na análise do grafo de colaborações entre autores de um programa de pós-graduação.

Nessa visualização, os nós que representam autores são coloridos de acordo com a linha de pesquisa de cada autor, com o objetivo de entender como se comportam as relações de colaboração entre autores de uma mesma linha de pesquisa dentro de um programa de pós-graduação. Em uma primeira análise, é possível observar que em grafos de co-autorias existe uma tendência de formação de núcleos de nós de mesma cor, indicando uma alta taxa de colaborações entre autores de uma mesma linha de pesquisa (ver figura 4.7). Esse comportamento está de acordo com as expectativas, visto que é natural que autores de uma mesma linha de pesquisa colaborem fortemente entre si.

Figura 5.3 – Visualização de autores centrais identificados no programa de computação da UFRGS.

Dentro de alguns desses núcleos, entretanto, foi possível notar que existem autores que se destacam em comparação com os demais. Isso porque esses autores ocupam uma posição central dentro do núcleo, possuem elevado número de produções acadêmicas e possuem alto número de conexões com os demais membros do núcleo de pesquisa. Esses

autores serão chamados de autores centrais.

Como exemplo, o programa de pós graduação em computação da UFRGS foi analisado. Na Figura 5.3, é possível observar autores centrais de 4 linhas de pesquisa distintas do programa. Para a linha de pesquisa de Computação de Alto Desempenho e Sistemas Distribuídos, identifica-se o docente Philippe Olivier Alexandre Navaux como um autor central, com elevado número de produções intelectuais e alta taxa de conexões com os demais autores da linha de pesquisa, o que segundo o modelo de densidade de conexões (detalhado na seção 4.2) indica que este autor está entre as colaborações mais relevantes de diversos outros autores. Outros exemplos de autores em posições centrais são: o docente Lisandro Zambenedetti Granville dentro da linha de pesquisa de Arquiteturas, Protocolos e Gerência de Redes e Serviço; os docentes Luciana Porcher Nedel e Anderson Maciel na linha de Interação Humano-computador, Realidade Virtual e Aumentada; e o docente Edson Prestes E Silva Junior na linha de Planejamento, Sistemas Multiagentes e Robótica.

6 CONSIDERAÇÕES FINAIS E TRABALHOS FUTUROS

A primeira proposta deste trabalho diz respeito à revisão da literatura existente relacionada aos temas de visualização de grafos de larga escala e análise de redes de co-autoria acadêmicas, bem como à exploração de ferramentas existentes para visualização, manipulação e armazenamento de grafos. Essa pesquisa, desenvolvida no capítulo 2 deste documento, foi realizada com o objetivo de buscar soluções para o problema da visualização dinâmica de uma rede de colaborações acadêmicas brasileira de larga escala. Embora a pesquisa não tenha revelado soluções existentes adequadas ao problema em questão, o processo serviu de base para a elaboração de uma solução completa envolvendo a geração de uma rede de colaborações acadêmicas de escala nacional a partir de dados da CAPES e a concepção da aplicação de visualização dinâmica VizColab, conforme exposto nos capítulos 3 e 4 deste trabalho.

Embora o trabalho tenha resultado em uma ferramenta completa e funcional para a exploração de uma rede brasileira de colaborações acadêmicas, contemplam-se expansões e melhorias que podem ser estudadas, testadas e eventualmente implementadas em trabalhos futuros. No que diz respeito à geração da rede de colaborações acadêmicas, este trabalho se limita aos dados disponibilizados nos conjuntos de dados da CAPES. Entende-se, entretanto, que o enriquecimento desses dados a partir da obtenção de informações de outras fontes de dados seria benéfico para o trabalho. Exemplos de informações adicionais que podem ser interessantes são: contagem de citações para produções intelectuais, índice-h de autores e conceito CAPES de programas de pós-graduação. Outro aspecto que considera-se digno ser estudado seria a concepção de um algoritmo de agrupamento de autores baseado em co-autorias frequentes, de forma que um alto número de co-autorias em comum entre dois indivíduos seria capaz de aumentar a confiança de que ambos se tratam de um mesmo autor.

No capítulo 5 analisou-se a usabilidade prática da ferramenta VizColab para a exploração da rede de colaborações e extração de *insights* a respeito de instituições de ensino, programas de pós-graduação e autores de produções intelectuais, bem como a respeito do estado das colaborações entre essas entidades. Dados os resultados positivos lá descritos, considera-se que este trabalho obteve êxito no cumprimento de suas propostas iniciais. Além deste documento, o trabalho ainda gerou um produto que, a partir desta data, estará a disposição da comunidade: a ferramenta VizColab[1].

[1] A ferramenta está disponível em http://vizcolab.inf.ufrgs.br/

REFERÊNCIAS

AGGRAWAL, N.; ARORA, A. Visualization, analysis and structural pattern infusion of dblp co-authorship network using gephi. In: **2016 2nd International Conference on Next Generation Computing Technologies (NGCT)**. [S.l.: s.n.], 2016. p. 494–500.

ASTURIANO, V. **Vasturiano/3D-force-graph: 3D force-directed graph component using threejs/webgl**. https://github.com/vasturiano/3d-force-graph/ (Acesso em 20 de Março de 2022), 2017.

CAPES. **Dados Abertos Capes**. https://dadosabertos.capes.gov.br/ (Acesso em 21 de Março de 2022), 2022.

CAPES. **Dados Abertos CAPES - Autor da Produção Intelectual de Programas de Pós-Graduação Stricto Sensu no Brasil**. https://dadosabertos.capes.gov.br/dataset/2017-a-2020-autor-da-producao-intelectual-de-programas-de-pos-graduacao-stricto-sensu (Acesso em 14 de Setembro de 2022), 2022.

CAPES. **Dados Abertos CAPES - Cursos da Pós-Graduação Stricto Sensu no Brasil**. https://dadosabertos.capes.gov.br/dataset/2017-a-2020-cursos-da-pos-graduacao-stricto-sensu-no-brasil (Acesso em 14 de Setembro de 2022), 2022.

CAPES. **Dados Abertos CAPES - Produção Intelectual de Pós-Graduação stricto sensu no Brasil**. https://dadosabertos.capes.gov.br/dataset/2017-a-2020-producao-intelectual-de-pos-graduacao-stricto-sensu-no-brasil (Acesso em 14 de Setembro de 2022), 2022.

CAPES. **Dados Abertos CAPES - Programas da Pós-Graduação Stricto Sensu no Brasil**. https://dadosabertos.capes.gov.br/dataset/2017-a-2020-programas-da-pos-graduacao-stricto-sensu-no-brasil (Acesso em 14 de Setembro de 2022), 2022.

CAVA, R.; FREITAS, C.; WINCKLER, M. Clustervis: visualizing nodes attributes in multivariate graphs. In: . [S.l.: s.n.], 2017. p. 174–179.

DEVI, N. M.; KASIREDDY, S. R. Graph analysis and visualization of social network big data. In: ____. **Social Network Forensics, Cyber Security, and Machine Learning**. Singapore: Springer Singapore, 2019. p. 93–104. ISBN 978-981-13-1456-8. Available from Internet: <https://doi.org/10.1007/978-981-13-1456-8_8>.

JUPYTER. **Project Jupyter**. https://jupyter.org/ (Acesso em 16 de Setembro de 2022), 2022.

KUMAR, S. Co-authorship networks: a review of the literature. **Aslib Journal of Information Management**, Emerald Group Publishing Limited, v. 67, n. 1, p. 55–73, Jan 2015. ISSN 2050-3806. Available from Internet: <https://doi.org/10.1108/AJIM-09-2014-0116>.

MDN. **Single Page Applications**. https://developer.mozilla.org/en-US/docs/Glossary/SPA (Acesso em 30 de Março de 2022), 2021.

META. **React - Uma biblioteca JavaScript para criar interfaces de usuário**. https://pt-br.reactjs.org/ (Acesso em 21 de Setembro de 2022), 2022.

MIYAMURA, H. N. et al. Adaptive view-dependent tree graph visualization. In: IEEE. **2011 Eighth International Joint Conference on Computer Science and Software Engineering (JCSSE)**. [S.l.], 2011. p. 187–192.

NEO4J. **Neo4j - Graph Data Platform**. https://neo4j.com/ (Acesso em 22 de Março de 2022), 2022.

NEO4J. **Neo4j - Product**. https://neo4j.com/product/ (Acesso em 28 de Março de 2022), 2022.

PANDAS. **Pandas - Python data analysis library**. https://pandas.pydata.org/ (Acesso em 16 de Setembro de 2022), 2022.

PERIANES-RODRÍGUEZ, A.; OLMEDA-GÓMEZ, C.; MOYA-ANEGÓN, F. Detecting, identifying and visualizing research groups in co-authorship networks. **Scientometrics**, v. 82, n. 2, p. 307–319, Feb 2010. ISSN 1588-2861. Available from Internet: <https://doi.org/10.1007/s11192-009-0040-z>.

PYTHON. **Python.org**. https://www.python.org/ (Acesso em 16 de Setembro de 2022), 2022.

SHI, L. et al. Himap: Adaptive visualization of large-scale online social networks. In: IEEE. **2009 IEEE Pacific Visualization Symposium**. [S.l.], 2009. p. 41–48.

SILVA JUNIOR, A. Nunes da et al. Analysis of co-authorship networks among brazilian graduate programs in computer science. **PLOS ONE**, Public Library of Science, v. 17, n. 1, p. 1–17, 01 2022. Available from Internet: <https://doi.org/10.1371/journal.pone.0261200>.

SPRITZER, A. et al. Towards a smooth design process for static communicative node-link diagrams. **Computer Graphics Forum**, v. 34, 06 2015.

SPRITZER, A.; FREITAS, C. Design and evaluation of magnetviz—a graph visualization tool. **IEEE transactions on visualization and computer graphics**, v. 18, p. 822–35, 06 2011.

VIéGAS, F.; DONATH, J. Social network visualization: Can we go beyond the graph. 01 2004.

WICKBOLDT, J. **PPGC Co-authorship Graph**. https://github.com/julianowick/ppgc-analysis (Acesso em 12 de Março de 2022), 2019.