



Universidade de Brasília

Instituto de Ciências Exatas
Departamento de Ciência da Computação

Uma investigação sobre ferramentas de visualização para análise de redes sociais em ecossistema educacional

Lucas de Lima Sousa Bezerra

Monografia apresentada como requisito parcial
para conclusão do Curso de Computação — Licenciatura

Orientador

Prof.a Dr.a Germana Menezes da Nóbrega

Brasília
2024



Uma investigação sobre ferramentas de visualização para análise de redes sociais em ecossistema educacional

Monografia apresentada como requisito parcial
para conclusão do Curso de Computação — Licenciatura

Prof. Dr. Jan Mendonça Correa Prof. Dr. Wilson Henrique Veneziano
CIC/UnB CIC/UnB

Prof. Dr. Jorge Henrique Cabral Fernandes
Coordenador do Curso de Computação — Licenciatura

Brasília, 12 de Setembro de 2024

CIP - Catalogação na Publicação

SD278i SOUSA BEZERRA, LUCAS DE LIMA .
Uma investigação sobre ferramentas de visualização para
análise de redes sociais em ecossistema educacional / LUCAS
DE LIMA SOUSA BEZERRA; orientador GERMANA MENEZES DA
NÓBREGA. -- Brasília, 2024.
60 p.

Monografia (Graduação - COMPUTAÇÃO) -- Universidade de
Brasília, 2024.

1. Visualização Gráfica. 2. Interações virtuais. 3. Redes
Sociais. I. DA NÓBREGA, GERMANA MENEZES, orient. II.
Título.

Dedicatória

Dedico esse trabalho primeiramente a Deus, que me deu forças, renovou minha mente e transformou o meu ser. Sem seu amor, nada disso seria possível. "Porque Deus amou o mundo de tal maneira que deu o seu Filho unigênito, para que todo aquele que nele crê não pereça, mas tenha a vida eterna." João 3:16. Também o dedico à minha família, em especial à minha mãe Danielle, ao meu pai Sinval, que sempre estiveram comigo, me dando todo o suporte que necessitei sem questionar, além de todo o aprendizado e amor que me proporcionaram. Esse laço familiar é um ponto de apoio o qual foi essencial na minha caminhada até a conclusão deste trabalho. Dedico esse trabalho à minha namorada, Vanessa, por todo carinho, amor, companheirismo e inspiração que me deu ao longo desses anos, além de ser a pessoa que sempre me deu força e apoio nos momentos mais difíceis dessa jornada. Dedico esse trabalho a toda minha família de Campina Grande, por me proporcionarem cuidado, amor, apoio e alegria, em especial minha avó Zélia, minha avó Marinalva, além das minhas queridas tias Gabrielle, Isabelle e Aline, e claro, meu tio La Martine. Sem vocês eu não seria quem eu sou hoje. Também agradeço aos meus sogros, Neuzanira e Israel, e sua família, pelos conselhos, pelo cuidado e por me fazerem sentir parte do hall familiar com todo carinho do mundo. Por fim, agradeço à minha igreja, minha segunda casa e família, por terem intercedido todos esses anos por mim e pela conclusão do curso. Vocês são os gigantes cujos ombros me permitiram enxergar mais longe. A todos, muito obrigado.

Agradecimentos

Agradeço à professora Germana Menezes, por confiar em mim nessa caminhada até aqui e por sua forma leve, profissional e de suporte na condução do trabalho, além de seu conhecimento espetacular na área. O departamento de Ciência da Computação está em excelentes mãos.

Resumo

A análise de redes sociais (SNA) tem se mostrado uma ferramenta poderosa para desvendar as complexas dinâmicas de interação em diversos contextos, incluindo o educacional. A representação gráfica de redes sociais permite visualizar padrões de conexão e identificar comunidades, influenciadores e estruturas ocultas no emaranhado da rede. No presente estudo, explora-se a aplicação da SNA em um ecossistema educacional, com o objetivo de avaliar diferentes ferramentas de visualização disponíveis para analisar as interações dos integrantes e extrair informações dessas interações. A revisão da literatura atesta que a utilização dessas ferramentas para extrair *insights* valiosos sobre os processos de ensino-aprendizagem, colaboração e socialização é muito eficaz. Os resultados da investigação forneceram apoio para a tomada de decisão de futuros projetistas do ecossistema, contribuindo para uma melhor análise e compreensão das interações sociais no ambiente acadêmico a partir de diferentes dados.

Palavras-chave: Redes Sociais, interações, visualização gráfica

Abstract

Social network analysis (SNA) has proven to be a powerful tool for unraveling the complex dynamics of interaction in various contexts, including education. The graphical representation of social networks enables the visualization of connection patterns and the identification of communities, influencers, and hidden structures within the network's intricacies. This study explores the application of SNA in an educational ecosystem, with the aim of evaluating different visualization tools available for analyzing interactions among members and extracting information from these interactions. The literature review confirms that the use of these tools to extract valuable insights into teaching and learning processes, collaboration, and socialization is highly effective. The results of the investigation provided support for decision-making by future ecosystem designers, contributing to a better analysis and understanding of social interactions in the academic environment based on diverse data.

Keywords: Social Networks, interactions, graphic visualization

Sumário

| | | |
|----------|---|-----------|
| 1 | Introdução | 1 |
| 1.1 | Contextualização | 1 |
| 1.2 | Motivação e Justificativa | 4 |
| 1.3 | Questão/Problema de Pesquisa | 5 |
| 1.4 | Objetivos | 6 |
| 1.5 | Metodologia | 6 |
| 1.6 | Estrutura deste documento | 7 |
| 2 | Fundamentação | 8 |
| 2.1 | Visualização da informação | 8 |
| 2.2 | Análise de Redes Sociais (SNA) e visualização | 10 |
| 3 | Panorama de ferramentas computacionais de SNA com visualização | 14 |
| 3.1 | Aplicação de Datasets | 14 |
| 3.2 | Características comuns a todas as ferramentas | 14 |
| 3.3 | Panorama de ferramentas computacionais de SNA com visualização | 17 |
| 3.3.1 | Cytoscape | 18 |
| 3.3.2 | NetworkX | 23 |
| 3.3.3 | Socnetv | 26 |
| 3.3.4 | Gephi | 30 |
| 3.3.5 | Graph-tool | 37 |
| 3.3.6 | Igraph | 40 |
| 4 | Um estudo comparativo de ferramentas à luz de um case | 45 |
| 4.1 | Case CICFriend e addon "Badges" para acreditação de mérito interacionista | 45 |
| 4.2 | O comparativo proposto | 46 |
| 4.3 | Redesenho do addon Badges incluindo visualização das interações | 48 |
| 5 | Conclusão | 53 |
| 5.1 | Objetivos alcançados | 53 |

| | |
|------------------------------------|-----------|
| 5.2 Considerações finais | 54 |
| 5.3 Trabalhos Futuros | 57 |
| Referências | 58 |

Lista de Figuras

| | | |
|------|--|----|
| 1.1 | Principais redes sociais entre os anos 2010 e 2020. | 2 |
| 1.2 | Arquitetura do ecossistema SmartUnB.ECOS. [1]. | 3 |
| 1.3 | Visualização do conjunto de dados Yaster Gene Interaction, na ferramenta Cytoscape v3.9.1. | 5 |
| 2.1 | Principais características para visualização das redes sociais do artigo [2]. . | 12 |
| 3.1 | Funcionalidades comuns em todas as ferramentas. A legenda mostra como o recurso está (ou pode estar) disponível na ferramenta em questão. As células marcadas com a cor verde significam que a funcionalidade está presente na ferramenta nativamente. As células marcadas em azul significam que a funcionalidade pode ser adicionada através de <i>plugins</i> disponibilizados na ferramenta. As células marcadas com a cor amarela significam que a funcionalidade pode ser adicionada através de implementação direta. A cor vermelha significa que o recurso não pode ser adicionado a ferramenta de forma alguma, porém, está ilustrada apenas para mostrar que todos as funcionalidades podem ser obtidos pelas ferramentas. | 15 |
| 3.2 | Tela inicial da ferramenta Cytoscape. Imagem retirada do manual da ferramenta [3]. | 19 |
| 3.3 | Modos de <i>layout</i> do Cytoscape [3] | 19 |
| 3.4 | Painéis de controle (a esquerda). | 20 |
| 3.5 | Visualização de grafos na aplicação <i>desktop</i> e no Cytoscape.js do mesmo grafo [3] | 22 |
| 3.6 | Esboço inicial de um grafo simples usando o NetworkX. | 24 |
| 3.7 | Interface <i>user friendly</i> do SocNetv [4]. | 27 |
| 3.8 | Construção de um grafo personalizado no SocNetv [4]. | 28 |
| 3.9 | Configurações gerais do SocNetv [4]. | 29 |
| 3.10 | Guia de Visão geral no Gephi [5]. | 31 |
| 3.11 | Laboratório de dados no Gephi [5]. | 32 |
| 3.12 | Área de pré visualização no Gephi [5]. | 33 |

| | | |
|------|---|----|
| 3.13 | Fácil aplicação dos filtros na ferramenta Gephi [5]. | 34 |
| 3.14 | Personalização dos elementos da rede no Gephi [5]. | 35 |
| 3.15 | Linha de tempo de uma rede no Gephi. | 35 |
| 3.16 | Exemplo de rede vista através do Gephi Lite [6]. | 36 |
| 3.17 | Grafo gerado com o Graph-tool usando dados da rede social Facebook [5]. . | 38 |
| 3.18 | Comparação entre Graph-tool, NetworkX e Igraph. Imagem retirada do site da ferramenta [7]. | 39 |
| 3.19 | Exemplo de visualização de comunidade no Igraph. | 41 |
| 3.20 | Exemplo de visualização de componentes conectados no Igraph [8]. | 41 |
| 3.21 | Figura retirada do artigo da ferramenta Igraph [9]. Refere-se a correta identificação de duas comunidades na rede <i>Zachary karate-club</i> . As linhas em vermelho são as arestas que se ligam entre as duas comunidades. Esse foi o resultado da aplicação do Algoritmo de localização de comunidade de Newman, através do Igraph. | 43 |
| 4.1 | Ferramentas para análise de redes sociais e seu comparativo com relação ao necessário para atuar no projeto. | 46 |
| 4.2 | Protótipo do Friendica com a análise de interações no menu do usuário. . . | 49 |
| 4.3 | Protótipo do Friendica com a análise de interações no menu superior. . . . | 49 |
| 4.4 | Protótipo do Friendica com a análise de interações no menu superior. . . . | 50 |
| 4.5 | Protótipo da área de interação em modo <i>fullscreen</i> na seção de configu- rações da ferramenta. | 51 |
| 4.6 | Protótipo da área de <i>badges</i> no menu superior de interações. | 52 |

Capítulo 1

Introdução

1.1 Contextualização

Ao longo dos anos, as redes sociais estão sendo cada vez mais exploradas, principalmente para fins sociais. As redes assumem diversos papéis e significados para os seus participantes. Orkut, Facebook, Instagram, TikTok e LinkedIn são apenas alguns dos principais exemplos de redes sociais, vide figura 1.1. Não é novidade que esses ambientes são um ponto de encontro de diversos públicos, que interagem entre si e, possivelmente, geram novas interações. Como define [10], as redes sociais representam “[...] um conjunto de participantes autônomos, unindo ideias e recursos em torno de valores e interesses compartilhados.”. As redes citadas anteriormente são exemplos de grupos que compartilham de um mesmo ponto focal de interesse e que interagem juntos para movimentação da rede. Uma definição simples, mas de fácil assimilação para o tema atual é também retratado em [10], onde a autora trás que as redes sociais são um sistema de nodos e elos. Como de fato é possível observar, as redes se formam de acordo com um agrupamento de nós (usuários), com um mesmo interesse em determinado tema.

Hoje em dia, esses ambientes tem se tornado cada vez mais específicos. Isso quer dizer que, uma vez que existe o conceito de comunidades dentro das redes sociais, essas comunidades, quando inseridas em um tema amplo, geram redes sociais baseadas nelas. Um exemplo disso é o conhecido LinkedIn, uma rede social profissional. Desse ponto de partida, é possível perceber que inúmeros outros nichos são capazes de gerar novos conglomerados de usuários com um mesmo objetivo. O ambiente que não deixou de ficar de fora foi o acadêmico. A inserção do ambiente educacional no mundo virtual contribui de maneira direta para aprendizagem e é um ponto de encontro entre a aprendizagem formal e a informal.

A união dos dois mundos, aprendizagem formal e informal, é um evento que contribui para a geração de interações. O trabalho [11], inserido no projeto de ecossistema virtual



Figura 1.1: Principais redes sociais entre os anos 2010 e 2020.

de aprendizagem, SmartUnB.ECOS [12], mostra como incorporar um mecanismo computacional que, a partir das interações dos discentes, prevê a emissão de um badge digital. Essas interações entre os discentes se tornam totalmente possíveis quando inseridas em um contexto educacional virtual.

Um ecossistema de aprendizagem é definido em [13] como um espaço onde indivíduos interagem, a partir de uma aprendizagem formal e informal contínua, com participação social e aprendizagem transformadora e significativa. Com essa definição, é fácil ver que redes sociais e ecossistemas estão ligados, visto que podem compartilhar características semelhantes com relação às interações, aprendizagem e participação social.

Como visto em [14] as mídias sociais tem ajudado os educadores a fazer das redes sociais locais de aprendizagem, com o objetivo de desenvolver comunidades de aprendizagem para colaboração e reflexão de ideias. Essa estrutura ajuda a fortalecer os laços do estudante com a comunidade e com os estudos, como também no desenvolvimento das ideias a medida que a discussão anda. O trabalho desenvolve uma rede estudantil com base no FaceBook e Moodle e, com base no que foi dito anteriormente, pode-se incluir essa ponte entre Moodle e FaceBook no mundo educacional. Essa inclusão pode permitir aos professores gerenciar as interações dos estudantes, dando o sentido educacional necessário e potencializando a inclusão de redes sociais no contexto educacional.

O presente trabalho está inserido em um projeto de ecossistema virtual de aprendizagem, denominado SmartUnB.ECOS [12], que é direcionado para o uso institucional e que a rede social implantada, explorada posteriormente, permite o estudo das interações entre usuários de todo o campus. Diferentemente da maioria das redes sociais que se popularizam ano após ano, que são redes centralizadas e com código fechado, a rede social que está no contexto do projeto é descentralizada e de código aberto, o que abre um leque de

possibilidades de personalização dentro de cada contexto acadêmico que esteja inserido. Como visto em [12], “A descentralização oferece ao usuário um grau de confiança quanto à exploração de seus dados [...]”, além disso, outra vantagem da descentralização é a distribuição da informação pelos nós, retratado em [15], pois, em redes sociais centralizadas, a remoção de um único nó (o nó central) destrói toda a comunicação dos nós subsequentes, enquanto na rede social descentralizada, por não existirem centros, a informação é disseminada por qualquer nó, deixando o poder e controle dos dados distribuído pela rede. A rede social a qual o projeto foi aplicado é o Friendica¹, que segue as características de descentralização e mantém seu código aberto, possibilitando a adaptação de recursos de acordo com o contexto ao qual esteja inserido.

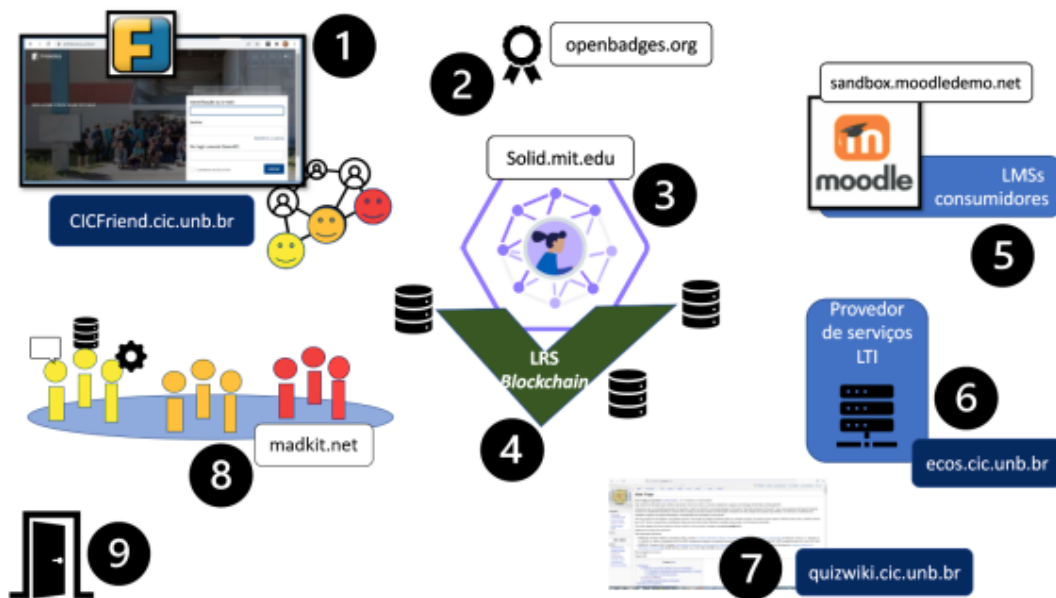


Figura 1.2: Arquitetura do ecossistema SmartUnB.ECOS. [1].

Com esses conceitos em mente, podemos observar a arquitetura do ecossistema SmartUnB.ECOS, vide Figura 1.2. Em [1] é possível verificar a referência que cada um dos elementos representa. A estrutura é descrita como se segue:

1. Representa a rede social descentralizada, implementada através da rede social Friendica.
2. Openbadges² é a plataforma de Gestão de *Badges* digitais, que são atribuídas aos usuários de acordo com suas atividades, de caráter informal.

¹<https://friendi.ca/>

²<https://openbadges.org/>

3. Um POD server para armazenamento de dados do usuário.
4. *Learning Records Store* utilizado com blockchain, para armazenar de forma distribuída os dados de registro de aprendizagem, com o devido aceite do usuário.
5. A exemplo da plataforma Moodle³, é a plataforma de gestão de aprendizagem tradicional, com suporte às disciplinas dos cursos.
6. Provedor *Learning Tools Interoperability* (LTI) para serviços específicos e ferramentas, para utilização em disciplinas.
7. QuizWiki⁴, é o repositório de Recursos Educacionais, que faz a indexação de forma semântica dos serviços dedicados advindos do provedor LTI.
8. Agentes artificiais autônomos encarregados da saúde acadêmica do corpo discente.
9. É a interface do ecossistema, com um mecanismo de personalização e de escape da bolha de filtro.

1.2 Motivação e Justificativa

No contexto do projeto SmartUnB.ECOS, com a rede social Friendica, tem-se um ambiente propício para o estudo de interações entre os usuários. Atualmente, porém, o projeto não conta com a possibilidade de visualizar graficamente as interações dos usuários (discentes, docentes e servidores). Como visto em [2] a visualização de redes sociais é um tema que está na pauta de empresas e da academia há alguns anos. Essa visualização “[...] permite, através de uma maneira natural, expressar a conectividade e facilitar a identificação de padrões na rede.” [2]. Em redes sociais, a quantidade de interações tende a ser muito grande devido ao número de usuários e as várias possibilidades de interagir com as ferramentas inseridas no contexto da rede. Sabendo disso, é fácil imaginar que tanto as redes quanto a sua relação de interações serão grandes. Essas relações ficam ocultas no grafo da rede e são passíveis de interpretação para que sua estrutura tenha um significado claro e também possa mostrar as tendências de comportamento. Um exemplo disso pode ser verificado na figura 1.3

Com a visualização clara destes dados, é possível fazer investigações com mais nitidez e fazer análises dos dados de forma mais profunda e detalhada. Como motivação para o desenvolvimento desse trabalho, temos o fato já citado anteriormente, de que a rede social implantada no ecossistema não tem uma ferramenta que possibilite a visualização gráfica

³<https://moodle.org>

⁴<http://quizwiki.cic.unb.br>

de interações e, por consequência disso, a análise de dados nas interações tem um grau de complexidade elevado.

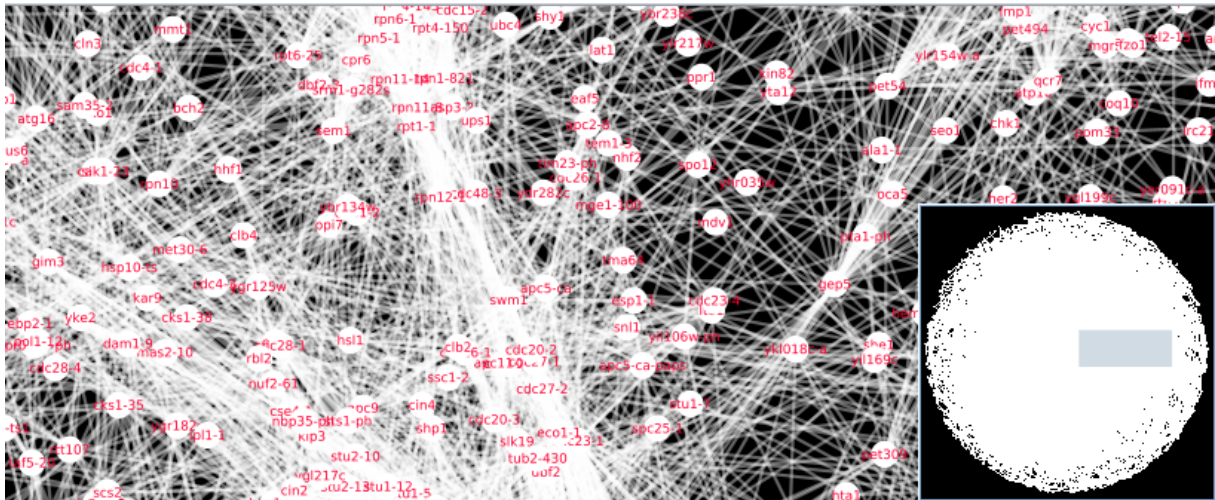


Figura 1.3: Visualização do conjunto de dados Yaster Gene Interaction, na ferramenta Cytoscape v3.9.1.

Além disso, como projetado para o ecossistema, a SmartUnB.ECOS terá um sistema de pontuação e recompensas como elemento de *gameificação* [11]. Os elementos básicos PBL (*points, badges and leaderboards*), são uma forma de atribuição de pontos para atividades entregues, medalhas (também chamadas de *badges*) obtidas por cumprir critérios, e organizar um *ranking* com as maiores pontuações. Para tanto, é necessário dar ao usuário uma forma visual de perceber suas interações dentro da rede.

1.3 Questão/Problema de Pesquisa

Com base na motivação e justificativa, o objetivo deste trabalho é resolver de forma prática a seguinte pergunta de pesquisa, com a ajuda da visualização de dados na análise de redes sociais: *Como dar suporte à escolha de futuros projetistas do ecossistema?*

Atualmente o projeto do Ecossistema conta com a rede social Friendica. A rede não conta com uma ferramenta de visualização de dados. Na arquitetura atual, o projeto necessita da visualização de interações, que seria incorporada através de ferramentas específicas para isso, assim como pode vir a necessitar dessas ferramentas em futuras atualizações e mudanças que possam ocorrer no ecossistema. De acordo com isso, a necessidade da visualização permanece, mas, outras funcionalidades incorporadas nessas ferramentas podem ser necessárias com o desenvolvimento do projeto. Dessa forma, será possível selecionar outras ferramentas de visualização que se adequam melhor na necessidade do ecossistema,

levando em consideração sua capacidade de incorporar outras funcionalidades ao ambiente em que foi inserida.

1.4 Objetivos

Objetivo Geral

Dando continuidade aos projetos desenvolvidos para compor o ecossistema virtual de aprendizagem SmartUnB.ECOS, o objetivo deste trabalho é avaliar e comparar ferramentas de visualização de redes sociais que tenham como características o código aberto, gratuidade para uso e que realizem a importação e exportação de dados, com o objetivo de auxiliar a escolha de uma ferramenta para a visualização de redes sociais no ecossistema educacional SmartUnB.ECOS, especificamente na instância Friendica (CICFriend).

Objetivos Específicos

- Realizar uma pesquisa para identificar e selecionar ferramentas de visualização de redes sociais que atendam aos critérios de código aberto, gratuidade e compatibilidade com os formatos de dados utilizados no ecossistema SmartUnB.ECOS.
- Avaliar as ferramentas selecionadas através de uma análise das funcionalidades, interfaces e capacidades de cada ferramenta, considerando as necessidades específicas do ecossistema SmartUnB.ECOS.
- Propor um redesenho do *addon badges*, integrando a ferramenta de visualização de redes sociais escolhida ao Friendica.

1.5 Metodologia

Para realização desse trabalho, uma pesquisa abrangente sobre as ferramentas de análise de redes sociais foi feita através de revisão de literatura e consulta a fontes de informação sobre essas ferramentas, como artigos, livros e sites especializados. Os repositórios de trabalhos científicos visitados foram:

- BRASNAM (Brazilian Workshop On Social Network Analysis And Mining)
- SBC (Sociedade Brasileira de Computação)
- IEEE (Institute of Electrical and Electronics Engineers)

Utilizaram-se as seguintes palavras-chave nas buscas nos repositórios: *Social Network Analysis*, *interactions*, *visualization*, *Social Network*. Os resultados foram filtrados por

artigos publicados nos últimos 5 anos e escritos em português ou inglês. A partir dos resultados da busca, foram selecionadas as ferramentas que atendiam aos seguintes critérios:

- A ferramenta precisa ter código aberto.
- A ferramenta precisa ser gratuita para uso.
- A ferramenta precisa ter uma licença para uso que permita o uso acadêmico, sem fins lucrativos e que permita alterações no produto.
- A ferramenta precisa ter uma visualização dos dados que seja razoavelmente fácil de se compreender.
- A ferramenta precisa ter a possibilidade de importar e exportar dados.
- A ferramenta precisa ser capaz de lidar com os principais formatos de dados disponíveis (JSON, XML, CSV ou GEXF).

As ferramentas selecionadas foram avaliadas com base em suas funcionalidades, documentação, interface e importação e exportação de dados. Para realizar a comparação entre as ferramentas, foram utilizados *datasets* públicos disponíveis na web, simulando cenários de análise de redes sociais. As 6 ferramentas que melhor se adaptaram aos critérios e aos objetivos da pesquisa foram selecionadas para análise mais detalhada.

Por fim, uma das ferramentas será indicada para ilustrar o seu uso em conjunto com o *addon badges*, dentro da rede social CICFriend.

1.6 Estrutura deste documento

O presente documento está estruturado em cinco capítulos principais. O primeiro capítulo é o de introdução, que contextualiza o projeto no qual o trabalho está inserido, juntamente com a rede social a qual o objeto de pesquisa será aplicado. Seguindo, temos a motivação e justificativa, a questão problema e o detalhamento da metodologia utilizada. No segundo capítulo, referente à fundamentação, é explicada a base para a visualização da informação e para análise de redes sociais se utilizando da visualização para compor a fase de análise dos dados. O terceiro capítulo apresenta um panorama de ferramentas para análise de redes sociais, onde são apresentadas algumas ferramentas para este fim. No capítulo 4, é apresentado um *case* para apresentar a proposta de incorporação da ferramenta na rede social utilizada no ecossistema do projeto. Por fim, no capítulo 5, será feita a conclusão do estudo, mostrando e elencando alguns objetos para futuros estudos.

Capítulo 2

Fundamentação

2.1 Visualização da informação

O artigo [16] é uma boa referência para entender o papel da visualização da informação, pois é possível compreender que a visualização tem o objetivo de auxiliar a análise de dados por meio de representações gráficas. Sendo assim, trabalhando com uma grande massa de dados, é possível enxergá-los e manipulá-los quando se está buscando alguma informação. No artigo, os autores falam que o processo de análise e interpretação visual dos dados é ainda mais significativo quando inserido em um contexto de ambiente virtual de aprendizagem. Nesse contexto, os autores citam o exemplo de professores que atuam nesses ambientes e podem querer melhorar suas aulas e a forma que a informação chega aos seus alunos. Para isso, a visualização da informação poderia ser usada para entender os tópicos de maior interesse dos alunos, estilos e preferências na hora de acessar os conteúdos.

O artigo [17] fala sobre visualização, que ao representar os dados de forma visual, tornando-os mais significativos e expressivos, é possível traduzi-los de forma a facilitar a exploração, análise e compreensão humana. O autor também fornece uma noção importante da visualização da informação, quando diz que uma pessoa pode estar analisando uma massa de dados, mas, pode ainda não ter em mente perguntas específicas para serem respondidas com a análise. A pessoa pode estar explorando os dados de forma a identificar questões, ponto no qual a visualização entra como forte aliado na identificação destes objetivos.

Em [16], os autores descrevem um processo de transformação de dados em uma representação visual no qual o usuário interage com os dados representados. O processo gira em torno de transformar os dados brutos, representá-los numa estrutura visual e exibi-la para o usuário. O usuário pode interagir com esses dados e alterá-los através da estrutura de visualização e, assim, pode aplicar o processo novamente e, possivelmente, gerar uma

estrutura diferente. Nesse processo de transformação de dados, são aplicadas técnicas de representação e de interação. Segundo o artigo, as técnicas de representação podem ser resumidas em:

- **Ortogonais:** É uma forma de visualização de dados que utiliza até três eixos (X, Y e Z) para organizar e exibir informações. Esses eixos ajudam a posicionar os dados de maneira que seja fácil de entender e interpretar. Além dos eixos, são usadas marcações, como pontos, linhas ou áreas, para representar os valores dos dados. Essa técnica é especialmente útil para dados tabulares com poucas variáveis, pois permite uma visualização clara e direta. Exemplos que usam essa técnica: Gráfico de barras, gráfico de setores, gráfico de bolas, dentre outros.
- **Multidimensionais:** É utilizada para visualizar dados tabulares que possuem mais de três variáveis. Quando as técnicas ortogonais não são suficientes para representar a complexidade dos dados, as técnicas multidimensionais são aplicadas. Elas permitem a visualização de múltiplas variáveis simultaneamente, facilitando a análise de dados complexos. Dois exemplos que usam essa técnica de visualização são o gráfico de coordenadas paralelas e o gráfico de radar.
- **Redes:** É utilizada para representar e analisar relacionamentos entre diferentes elementos. Esses relacionamentos podem ser visualizados de várias formas, sendo as mais comuns os grafos e as matrizes. Alguns exemplos de aplicação são redes de computadores e modelos de entidade-relacionamento.
- **Árvores:** As árvores são usadas para representar relações hierárquicas ou de composição entre elementos. Diferente das redes, que podem ter relações mais complexas e variadas, as árvores focam em estruturas hierárquicas claras, como diretórios de arquivos ou árvores genealógicas. Alguns exemplos de técnicas da aplicação de árvores são as *treemaps* e árvores hiperbólicas.

As técnicas de interação são basicamente aquelas que são disponibilizadas para que o usuário possa fazer as manipulações nos dados, para representá-los de maneira diferente. Essas técnicas são:

- **Seleção:** Permite que o usuário selecione itens de interesse. Particularmente útil quando muitos itens são representados, pois facilita o acompanhamento e a distinção dos itens selecionados em meio aos demais.
- **Exploração:** Permite que o usuário examine diferentes subconjuntos de dados, mostrando novos itens e escondendo outros. Isso ajuda a focar em partes específicas dos dados sem perder a visão geral. Um exemplo é a navegação por páginas em um conjunto de dados grande, onde apenas uma parte dos dados é exibida por vez.

- **Reconfiguração:** Fornece ao usuário diferentes perspectivas dos dados representados através de alterações na organização dos itens, como alinhamento e ordem. Por exemplo, reordenar colunas em uma tabela para comparar diferentes métricas lado a lado.
- **Codificação:** Possibilita que o usuário altere a representação visual dos dados, como cor, tamanho, forma e estrutura visual. Isso ajuda a destacar diferentes aspectos dos dados. Um exemplo é mudar a cor dos nós de um grafo para representar um conjunto específico de dados.
- **Abstração:** Habilita o usuário a ajustar o nível de abstração dos dados representados, permitindo trocar de uma visão geral para uma visão detalhada e vice-versa. Por exemplo, um mapa interativo que permite *zoom* para ver detalhes específicos ou afastar para ver a visão geral.
- **Filtragem:** Permite que o usuário altere o conjunto de dados representado com base em condições específicas. Por exemplo, filtrar uma tabela de vendas para mostrar apenas as vendas do mês de dezembro ou de uma seção específica, como Informática.
- **Conexão:** Permite que o usuário destaque associações e relações entre itens representados ou mostre itens escondidos relevantes a outro item. Por exemplo, ao clicar em um nó em um grafo, destacar todos os nós conectados a ele para mostrar as relações.

Com isso, é possível perceber que a visualização de dados desempenha um papel fundamental no tratamento e na transformação dos dados para apresentação. As técnicas de visualização e interação trabalham em conjunto para transformar dados brutos em informações compreensíveis e úteis. Juntas, essas técnicas não apenas melhoram a compreensão dos dados, mas também potencializam a tomada de decisões informadas e estratégicas, permitindo que os usuários extraiam *insights* valiosos de grandes conjuntos de dados.

2.2 Análise de Redes Sociais (SNA) e visualização

Um dos principais trabalhos sobre a temática de visualização de dados em redes sociais, o artigo "*Visualização Interativa de Redes Sociais: um Estudo de Caso em Redes de Colaboração Científica*"[2], faz uma introdução da popularização das redes sociais ao longo dos anos com a facilidade do acesso a internet, o que possibilitou que vários estudos fossem aprofundados para compreender as relações sociais e como elas são estabelecidas. Esses estudos foram e continuam sendo muito importantes para identificação de padrões na

rede, temas trabalhados tanto na análise de dados quanto na visualização de informações. No desenvolvimento do trabalho, a hipótese a ser provada é a do “[...] uso de operações de conjuntos, aplicadas na construção gradual do grafo, podendo evitar mudanças bruscas de *layout*, melhorando a construção e manutenção do mapa mental da rede social pelo usuário”. As operações que são aplicadas nos grafos de interação são: união, soma, produto cartesiano e composição. Além disso, duas operações unárias também foram abordadas, sendo elas: contração de vértice e inclusão de vértice. O trabalho faz um apanhado com as principais ferramentas de visualização de interações em redes sociais da época para em seguida analisar as principais características para se fazer a visualização de redes sociais, bem como sua disponibilidade em cada uma das ferramentas analisadas. A Figura 2.1 mostra a relação das ferramentas e da disponibilidade das características em cada uma delas. As características listadas são as que foram julgadas mais importantes para visualização de redes sociais, sendo elas:

- **Zoom:** Capacidade de aproximar e afastar sem perda de foco.
- **Acesso ao conteúdo:** Capacidade de verificar os atributos e valores de cada nó e aresta.
- **Origem dos dados:** Permite acesso aos dados originais usados para a construir o grafo.
- **Consulta no grafo:** Capacidade de consultar os nós a partir dos valor e conteúdo dos atributos.
- **Filtro:** Capacidade de estabelecer critérios para exibir um conjunto de nós e ocultar outros.
- **Expandir:** Capacidade de modificar a estrutura visível do grafo, mostrando ou ocultando partes específicas. Um exemplo para ilustrar é o de abrir ou fechar um diretório para visualizar ou esconder seu conteúdo.
- **Agrupar:** Capacidade de agrupar nós. Esse agrupamento pode ou não seguir um critério específico.
- **Representação dos dados:** Capacidade de definir a estrutura utilizada para a representação gráfica da rede.

Essas características são essenciais para realizar uma análise de redes sociais de maneira a extrair o máximo de dados da rede em questão. O quadro da Figura 2.1 permite realizar uma rápida comparação entre cada uma das ferramentas selecionadas, visando uma tomada de decisão mais rápida ao verificar se a ferramenta possui as características necessárias para se trabalhar com análise.

A ferramenta scripLattes foi a que apresentou menor número de recursos, mas, possuía um grande potencial de evolução, como retratado no trabalho. Além disso, todas as ferramentas continuaram em desenvolvimento e evolução, contudo, a ScripLattes tinha o interesse do público pesquisador brasileiro para desenvolver suas funcionalidades e também para visualizar e analisar dados de coautoria. À medida que o ScriptLattes é melhorado, ele pode facilitar a colaboração e o compartilhamento de conhecimento entre pesquisadores brasileiros. Isso ocorre porque a ferramenta torna mais fácil analisar e entender os dados de coautoria. A análise comparativa realizada no artigo levou ao desenvolvimento de uma nova funcionalidade para o ScriptLattes, que já está disponível. Essa funcionalidade realiza a união de grafos para explorar redes de coautorias. Este artigo trabalha a visualização de redes de colaboração científica com o intuito de diminuir a complexidade da representação gráfica, mantendo a visualização eficiente relacionada às interações dos indivíduos.

| Ferramenta | Zoom | Acesso Conteúdo | Origem dos dados | Consulta no grafo | Filtra grafo | Expand grafo | Agrupar | Representa |
|-------------------|-------------|----------------------------|---------------------------------|------------------------------|-------------------------|-------------------------|----------------|-------------------|
| Vizster | Sim | Sim | Sim | Sim | Sim | Sim | Sim | grafo |
| MagnetViz | Sim | Sim | Não | Sim | Sim | Não | Sim | grafo |
| TouchGraph | Sim | Não | Sim | Não | Não | Sim | Sim | grafo |
| MatrixExplorer | Sim | Sim | Não | Sim | Sim | Não | Sim | matriz e grafo |
| scriptLattes | Não | Sim | Sim | Não | Não | Não | Não | grafo |

Figura 2.1: Principais características para visualização das redes sociais do artigo [2].

O artigo [18] fala a respeito da bolha de filtro, que se refere a quando um usuário é inserido em um fluxo de dados que são personalizados de acordo com o seu interesse. O trabalho aborda o momento em que uma pessoa tem um conteúdo filtrado, pois não está de acordo com o tipo de dados visualizado por ele, porém, se esse conteúdo é de interesse comum das pessoas, seria algo grave, colocando o usuário em um “mundo isolado”. Nesse trabalho, os autores abordam a sobrecarga de dados gerada por grandes redes sociais, o que ocasiona uma tendência a filtrar conteúdo irrelevante para os usuários. Essa tendência em filtrar, faz com que o usuário seja inserido numa “bolha de filtro”, onde apenas o conteúdo relevante a ele será mostrado. A ideia abordada no artigo é que o usuário esteja consciente sobre o mecanismo de filtragem e assuma o controle sobre ele, fazendo com que seja possível pesquisar nos itens filtrados, fazendo o modelo de usuário resultante do algoritmo da rede possa ser mudado caso o usuário deseje. Porém, dessa maneira, o usuário será reinserido no problema da sobrecarga de dados. Logo, a ideia é desenvolver uma técnica de visualização que possa ser integrada sem adicionar sobrecarga de dados.

A bolha desenvolvida separa os conteúdos de dentro e de fora da bolha. O conteúdo de dentro é mostrado e os de fora não, porém, na visualização de dados eles aparecem. Então, existem duas possibilidades de visualização para os dados personalizados:

1. Visualização com foco nos amigos: De maneira análoga à visualização com foco nas categorias, aqui os dados são exibidos com base nas postagens dos amigos.
2. Visualização com foco nas categorias semânticas dos dados sociais originários no OSN: o mecanismo seleciona categorias (Ex.: Saúde, moda, esportes, etc) que são de interesse mútuo entre os amigos do usuário e o próprio usuário e filtra aqueles que não são de interesse mútuo. No entanto, é crucial garantir que essa seleção respeite a privacidade dos usuários, evitando a exposição de dados que não foram explicitamente compartilhados para esse fim. A ideia é que o usuário tenha essa visualização e que possa escolher entre ver ou não um conteúdo de dentro da bolha ou de fora a partir das categorias, ou, a partir dos amigos, sempre respeitando as configurações de privacidade estabelecidas.

Essas visualizações mostram que existe uma relação entre a categoria das postagens e sua origem. A proposta é que o usuário tenha controle sobre o que deseja visualizar, podendo optar por conteúdos de dentro ou fora da bolha, com base nas categorias ou nas postagens dos amigos, sem comprometer a privacidade dos dados.

Outra referência de grande valor e que trabalha a visualização de interações em rede social estudantil é [14]. O estudo mostra a importância do conceito de *Social Student Relationship Management* (Social SRM). O trabalho implanta uma mistura do Moodle com o Facebook (o chamado EduBridge Social), para melhorar a interatividade e as discussões e criar um ambiente privilegiado para discussões. O ambiente educacional deve ter qualidade na comunicação, pois isso impacta diretamente na eficiência da aprendizagem. Quando essa qualidade na comunicação se combina com as redes sociais, as possibilidades de aprendizagem são infinitas.

Em [19], os autores investigam uma abordagem computacional para identificar líderes de aprendizagem através da aprendizagem colaborativa usando como métricas o *indegree*, *outdegree* e *betweenness*. Contextualizados no artigo, *indegree* é o número de mensagens recebidas por um aluno, *outdegree* é o número de mensagens enviadas por um aluno e *betweenness* é a extensão em que um aluno controla a comunicação entre dois outros alunos em uma comunidade. Assim, essas métricas ajudam a analisar a dinâmica social em discussões online e auxiliam na identificação de líderes de aprendizagem. Os alunos foram categorizados em diferentes papéis, como facilitador completo, facilitador transacional e facilitador atraente, com base nas métricas de SNA. A identificação de líderes de aprendizagem variou ao longo das semanas, refletindo a dinâmica e a interação situacional dos alunos na comunidade online.

Capítulo 3





































Panorama de ferramentas computacionais de SNA com visualização

3.1 Aplicação de Datasets

Para tornar possível ver o funcionamento pleno das ferramentas, é necessário a utilização de dados para visualizar como eles são estruturados. Algumas das ferramentas já tinham dados disponíveis para esse fim, não sendo necessário coletar de Datasets externos, como é o caso da Cytoscape, Jung, Social Network Visualizer, dentre outras. Quando a ferramenta não contava com um conjunto de dados para teste, foi necessário aplicar *datasets* externos. *Datasets* são conjuntos de dados reunidos de uma ou mais fontes. O conjunto de dados usado nesse trabalho foi o *Stanford Large Network Dataset Collection* [20], que fornece uma lista com dados públicos de diversas fontes. A fonte escolhida para o caso foi a *Social circles* da rede social Twitter [21].

3.2 Características comuns a todas as ferramentas

Navegar pelo cenário diversificado de ferramentas de análise de rede e abranger todas as características de cada uma pode ser cansativo. Devido a natureza das ferramentas e por terem sido desenvolvidas com o mesmo intuito, algumas funcionalidades e recursos são comuns a todas as ferramentas. Devido a isso, para que a descrição de cada ferramenta não fique repetitiva, os recursos compartilhados serão explicitados adiante. A Figura 3.1 a seguir mostra a relação de recursos comuns para as ferramentas e como se faz possível a sua adição à ferramenta.

| Ferramenta | Detectar comunidades | Análise de dados | Encontrar o menor caminho | Cálculo de métricas | Desenvolvimento de plugins e scripts | Visualização e personalização |
|------------|---|---|---|---|---|---|
| Cytoscape |  |  |  |  |  |  |
| NetworkX |  |  |  |  |  |  |
| SocnetV |  |  |  |  |  |  |
| Gephi |  |  |  |  |  |  |
| Graph-tool |  |  |  |  |  |  |
| Igraph |  |  |  |  |  |  |





Legenda:  Tem nativo
  Não tem nativo, mas tem em plugins
  Não tem nativo, mas pode ser implementado
  Não tem nativo e não pode ser adicionado

Figura 3.1: Funcionalidades comuns em todas as ferramentas. A legenda mostra como o recurso está (ou pode estar) disponível na ferramenta em questão. As células marcadas com a cor verde significam que a funcionalidade está presente na ferramenta nativamente. As células marcadas em azul significam que a funcionalidade pode ser adicionada através de *plugins* disponibilizados na ferramenta. As células marcadas com a cor amarela significam que a funcionalidade pode ser adicionada através de implementação direta. A cor vermelha significa que o recurso não pode ser adicionado a ferramenta de forma alguma, porém, está ilustrada apenas para mostrar que todos as funcionalidades podem ser obtidos pelas ferramentas.

Alguns desses recursos não estão presentes nativamente na ferramenta, mas é possível adicionar através de *plugins* e *scripts*. Quando não existe a opção de adicionar por meio de *plugins*, algumas ferramentas possibilitam a adição dos recursos através das linguagens de programação relacionada, sendo possível então implementar a funcionalidade em questão. De início é possível notar que apenas duas ferramentas possuem todos os recursos nativamente (NetworkX e Igraph). Apesar disso, é possível obter o recurso, na maioria dos casos, através da loja de aplicativos e *scripts* de *software*. Apenas no caso dos algoritmos de *short path* é necessária a implementação na linguagem nativa da ferramenta.

Uma funcionalidade muito importante para a análise de redes sociais é a detecção de comunidades. Essa é uma técnica crucial para identificar *clusters* de nós dentro de uma rede que apresentam uma maior densidade de conexões entre si em comparação com o resto da rede. Esta funcionalidade permite descobrir grupos ou comunidades ocultas dentro da rede, representando potencialmente categorias temáticas, unidades funcionais ou entidades em interação.

Outro recurso essencial para SNA é a análise de dados. Ela permite aos ir além da mera visualização, abrangendo o cálculo de várias métricas de rede que quantificam propriedades estruturais específicas da rede. Os exemplos incluem métricas como:

- **Centralidade de grau:** Representa a contagem de conexões diretas de um nó, indicando a popularidade ou atividade de um nó. Um nó com alta centralidade de grau tem muitas conexões diretas e pode ser considerado influente ou central na rede.
- **Centralidade de intermediação:** Representa a importância de um nó na conexão de diferentes partes da rede. Essa métrica mede quantas vezes um nó atua como ponte em caminhos mais curtos entre outros nós. Um nó com alta centralidade de intermediação é crucial para a comunicação e o fluxo de informações na rede.
- **Detecção de anomalias:** São padrões inesperados ou incomuns na estrutura da rede. Essa métrica pode identificar nós ou conexões que se desviam do comportamento esperado, indicando possíveis influenciadores, caminhos críticos ou interrupções.
- **Distribuição de Grau:** Representa a distribuição das conexões entre os nós da rede. Essa métrica auxilia a entender a estrutura geral da rede, como a presença de *hubs* (nós altamente conectados) ou a existência de uma rede homogênea.

Essas métricas não mostram apenas médias ou distribuições, mas fornecem uma visão detalhada sobre a posição e o papel de cada nó dentro da rede. Elas ajudam a identificar nós influentes, entender a dinâmica de comunicação e detectar possíveis vulnerabilidades ou pontos de falha na rede.

A detecção do menor caminho, foi um recurso inserido na tabela para ilustrar que é possível a implementação de algumas funções através de linguagens de programação, como é o caso do Graph-tool, Socnetv e Networkit. Por exemplo, a Networkit, construída em Python, possui funções nativas que podem ser chamadas diretamente com a linguagem para fazer a construção do algoritmo de menor caminho. As demais funcionalidades também fornecem recursos e o ambiente necessário para a implementação das funcionalidades desejadas, no caso, a detecção do menor caminho.

Além disso, as ferramentas possuem a habilidade de realizar cálculo de métricas. Existem várias métricas que podem ser extraídas da rede através de algumas funcionalidades. Entre as principais métricas, temos:

- **Medidas de centralidade:** Essas métricas visam quantificar a importância relativa ou influência de nós individuais na rede. Dentre elas pode-se citar a *degree centrality*,

que mede o número de conexões (arestas) que um nó possui com outros nós, *Betweenness centrality*, que mede a frequência com que um nó se encontra no caminho mais curto entre outros pares de nós, indicando seu papel potencial como ponte ou guardião do fluxo de informações. Podemos citar ainda o *Closeness centrality*, que mede o quão próximo um nó está de todos os outros nós da rede, apontando sua facilidade de acesso a informações.

- Coeficiente de agrupamento: mede a densidade de conexões dentro de grupos de nós, indicando a tendência dos nós de se conectarem com outros dentro de sua vizinhança imediata.
- Distribuição do comprimento do caminho: analisa a distribuição do número de etapas necessárias para percorrer entre quaisquer dois nós escolhidos aleatoriamente, fornecendo insights sobre a conectividade geral da rede.

Finalmente, a base das ferramentas de análise de rede é a visualização de dados. Essas funcionalidades permitem representar visualmente a estrutura da rede, facilitando a exploração de relacionamentos, padrões e anomalias nos dados. Ao empregar vários algoritmos de layout e representações personalizáveis de nós/bordas, é possível comunicar *insights* de maneira eficaz e melhorar a compreensão da rede sob investigação.

Esses recursos comuns destacam as funcionalidades compartilhadas que vão auxiliar o trabalho no ecossistema para atribuição das *badges* digitais e futuros recursos implementados na rede.

3.3 Panorama de ferramentas computacionais de SNA com visualização

Nesta seção serão apresentadas as ferramentas para análise de redes sociais, dando destaque também para os recursos exclusivos ou que não estão presentes em todas as ferramentas. Um total de 6 ferramentas foram escolhidas para serem analisadas e inseridas como opção para o ecossistema SmartUnB.ECOS. A escolha dessas ferramentas não quer dizer que outras não podem ser inseridas ou consideradas, mas levando em consideração os aspectos apresentados no capítulo 1, essas ferramentas apresentam as características necessárias para serem inseridas no projeto. Caso futuramente as especificações do projeto sofram alterações, uma nova análise pode ser feita para abranger mais ferramentas que possam contemplar melhor a necessidade do ecossistema.

3.3.1 Cytoscape

O Cytoscape é uma ferramenta de visualização e análise de redes e grafos que permite aos usuários visualizar, explorar e analisar dados de redes complexas. Ele é um *software* de código aberto e gratuito, disponível para download através de seu código no repositório Github [22] e sob a licença BSD 2-Clause. Além disso, a ferramenta oferece uma grande variedade de funcionalidades e recursos avançados para análise de redes e grafos, como o suporte a diferentes tipos de dados, algoritmos de análise avançada e visualização de dados em 3D. Ele permite ao usuário importar, visualizar e manipular redes complexas, bem como aplicar diversas ferramentas para análise. Além disso, possui uma comunidade ativa de usuários e desenvolvedores, garantindo suporte e atualizações. O Cytoscape é compatível com Windows, MacOS e Linux e pode ser baixado gratuitamente através do site oficial [3]. Possui API (*Application Programming Interface*) aberta para desenvolvimento de aplicativos (ou *plugins*) em Java. O desenvolvimento de aplicativos pela comunidade é incentivado. Os aplicativos representam recursos adicionais para integrar o Cytoscape. Essa ferramenta ainda conta com uma versão *web* chamada Cytoscape.js, que é uma biblioteca JavaScript para visualização e análise de rede. Apesar de levar o mesmo nome da ferramenta original para *desktop*, o Cytoscape.js não é uma versão completa.

Interface

O Cytoscape possui uma interface de usuário especificamente projetada para visualização e análise de redes. Falando um pouco sobre como se divide a interface, temos primeiramente a área central, que é um grande espaço de trabalho onde a visualização da rede será exibida. Este espaço permite ampliar, deslocar e manipular o *layout* da rede.

Alguns menus estão localizados na parte superior da janela, os menus fornecem acesso a diversas funcionalidades, como gerenciamento de arquivos (abrir, salvar, exportar redes), ferramentas de edição de rede (adicionar/remover nós e arestas), algoritmos de *layout*, ferramentas de análise e definição de preferências. Além disso, o Cytoscape possui barras de ferramentas, que ficam horizontalmente abaixo dos menus e oferecem acesso rápido a funções usadas com frequência, como adicionar nós ou selecionar elementos de rede.

Além disso, a ferramenta possui 4 estilos para *layout* e organização dos dados. São eles: cascata, GRID, *side by side* e vertical. Uma prévia de cada estilo pode ser visto na figura 3.3.

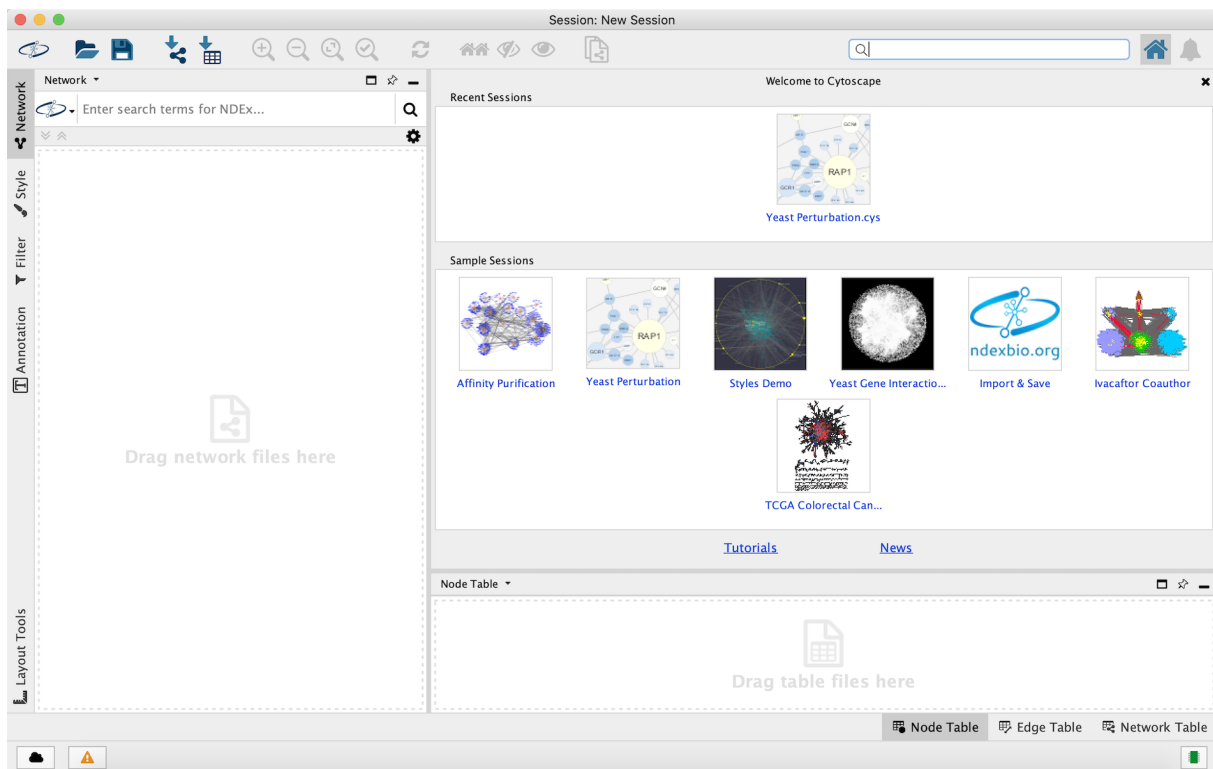


Figura 3.2: Tela inicial da ferramenta Cytoscape. Imagem retirada do manual da ferramenta [3].

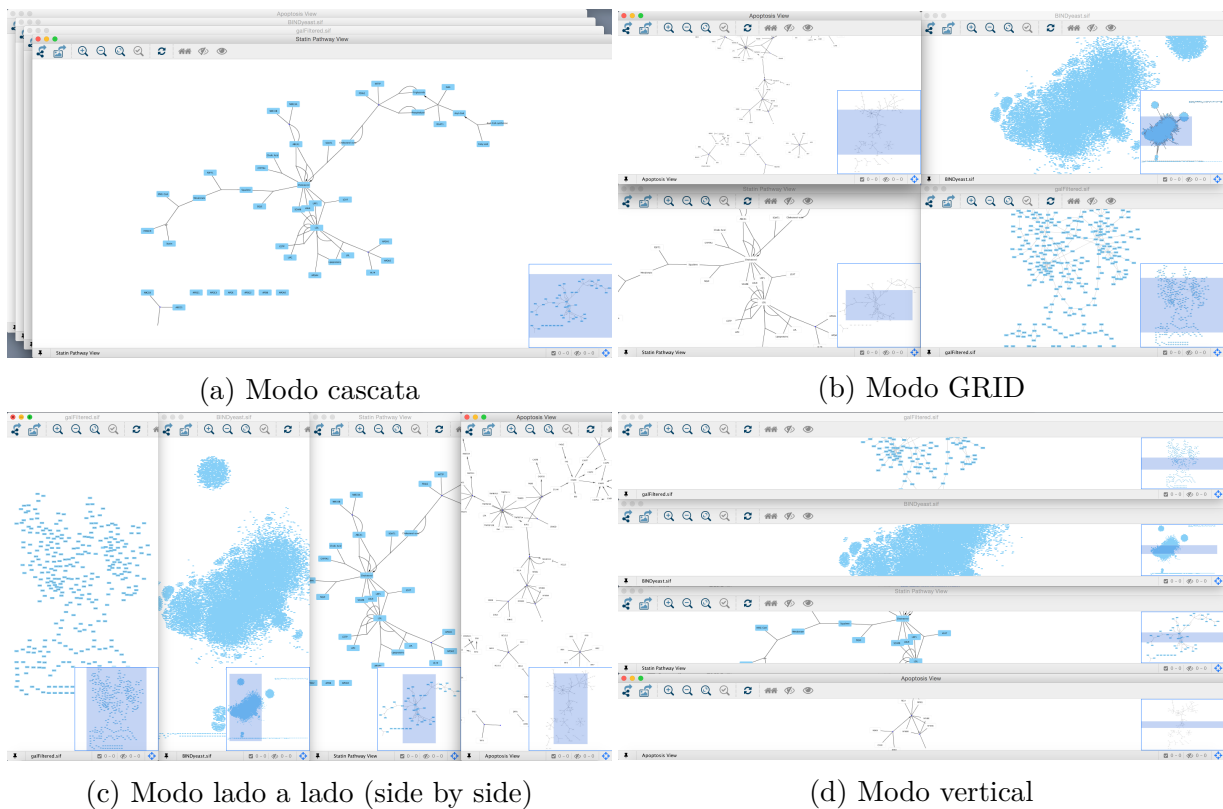


Figura 3.3: Modos de *layout* do Cytoscape [3]

The screenshot displays a metabolic pathway analysis software interface. The top bar shows the session path: `/Users/kono/Desktop/cy32-release2.cys`. The main window is titled "Amino sugar and nucleotide sugar metabolism [ko0520]".

Control Panel:

- Network:** Amino sugar and nucleotide sugar metabolism [k] (275(0) nodes, 275(0) edges).
- Style:** Select

Tool Panel:

- Align:** Buttons for alignment and distribution.
- Distribute:** Buttons for distribution and scaling.
- Stack:** Buttons for stacking and rotating.

Main Diagram:

The diagram illustrates the metabolic pathways for amino sugar and nucleotide sugar metabolism. Key components include:

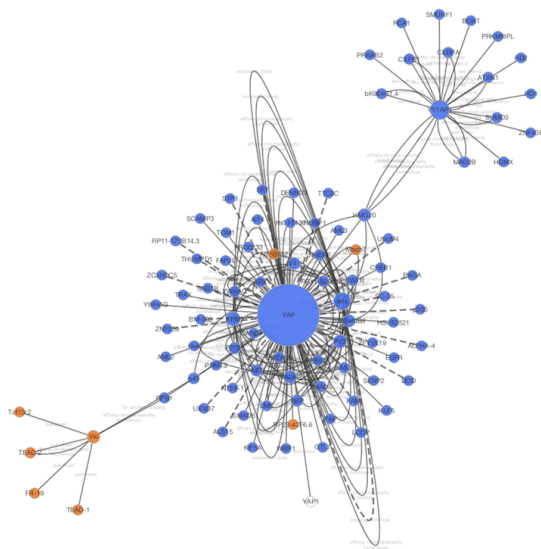
- Nodes:** Represented by circles with IDs (e.g., K01193, K01194, K01195, K01196, K01197, K01198, K01199, K01200, K01201, K01202, K01203, K01204, K01205, K01206, K01207, K01208, K01209, K01210, K01211, K01212, K01213, K01214, K01215, K01216, K01217, K01218, K01219, K01220, K01221, K01222, K01223, K01224, K01225, K01226, K01227, K01228, K01229, K01230, K01231, K01232, K01233, K01234, K01235, K01236, K01237, K01238, K01239, K01240, K01241, K01242, K01243, K01244, K01245, K01246, K01247, K01248, K01249, K01250, K01251, K01252, K01253, K01254, K01255, K01256, K01257, K01258, K01259, K01260, K01261, K01262, K01263, K01264, K01265, K01266, K01267, K01268, K01269, K01270, K01271, K01272, K01273, K01274, K01275, K01276, K01277, K01278, K01279, K01280, K01281, K01282, K01283, K01284, K01285, K01286, K01287, K01288, K01289, K01290, K01291, K01292, K01293, K01294, K01295, K01296, K01297, K01298, K01299, K01300, K01301, K01302, K01303, K01304, K01305, K01306, K01307, K01308, K01309, K01310, K01311, K01312, K01313, K01314, K01315, K01316, K01317, K01318, K01319, K01320, K01321, K01322, K01323, K01324, K01325, K01326, K01327, K01328, K01329, K01330, K01331, K01332, K01333, K01334, K01335, K01336, K01337, K01338, K01339, K01340, K01341, K01342, K01343, K01344, K01345, K01346, K01347, K01348, K01349, K01350, K01351, K01352, K01353, K01354, K01355, K01356, K01357, K01358, K01359, K01360, K01361, K01362, K01363, K01364, K01365, K01366, K01367, K01368, K01369, K01370, K01371, K01372, K01373, K01374, K01375, K01376, K01377, K01378, K01379, K01380, K01381, K01382, K01383, K01384, K01385, K01386, K01387, K01388, K01389, K01390, K01391, K01392, K01393, K01394, K01395, K01396, K01397, K01398, K01399, K01400, K01401, K01402, K01403, K01404, K01405, K01406, K01407, K01408, K01409, K01410, K01411, K01412, K01413, K01414, K01415, K01416, K01417, K01418, K01419, K01420, K01421, K01422, K01423, K01424, K01425, K01426, K01427, K01428, K01429, K01430, K01431, K01432, K01433, K01434, K01435, K01436, K01437, K01438, K01439, K01440, K01441, K01442, K01443, K01444, K01445, K01446, K01447, K01448, K01449, K01450, K01451, K01452, K01453, K01454, K01455, K01456, K01457, K01458, K01459, K01460, K01461, K01462, K01463, K01464, K01465, K01466, K01467, K01468, K01469, K01470, K01471, K01472, K01473, K01474, K01475, K01476, K01477, K01478, K01479, K01480, K01481, K01482, K01483, K01484, K01485, K01486, K01487, K01488, K01489, K01490, K01491, K01492, K01493, K01494, K01495, K01496, K01497, K01498, K01499, K01500, K01501, K01502, K01503, K01504, K01505, K01506, K01507, K01508, K01509, K01510, K01511, K01512, K01513, K01514, K01515, K01516, K01517, K01518, K01519, K01520, K01521, K01522, K01523, K01524, K01525, K01526, K01527, K01528, K01529, K01530, K01531, K01532, K01533, K01534, K01535, K01536, K01537, K01538, K01539, K01540, K01541, K01542, K01543, K01544, K01545, K01546, K01547, K01548, K01549, K01550, K01551, K01552, K01553, K01554, K01555, K01556, K01557, K01558, K01559, K01560, K01561, K01562, K01563, K01564, K01565, K01566, K01567, K01568, K01569, K01570, K01571, K01572, K01573, K01574, K01575, K01576, K01577, K01578, K01579, K01580, K01581, K01582, K01583, K01584, K01585, K01586, K01587, K01588, K01589, K01590, K01591, K01592, K01593, K01594, K01595, K01596, K01597, K01598, K01599, K01600, K01601, K01602, K01603, K01604, K01605, K01606, K01607, K01608, K01609, K01610, K01611, K01612, K01613, K01614, K01615, K01616, K01617, K01618, K01619, K01620, K01621, K01622, K01623, K01624, K01625, K01626, K01627, K01628, K01629, K01630, K01631, K01632, K01633, K01634, K01635, K01636, K01637, K01638, K01639, K01640, K01641, K01642, K01643, K01644, K01645, K01646, K01647, K01648, K01649, K01650, K01651, K01652, K01653, K01654, K01655, K01656, K01657, K01658, K01659, K01660, K01661, K01662, K01663, K01664, K01665, K01666, K01667, K01668, K01669, K01670, K01671, K01672, K01673, K01674, K01675, K01676, K01677, K01678, K01679, K01680, K01681, K01682, K01683, K01684, K01685, K01686, K01687, K01688, K01689, K01690, K01691, K01692, K01693, K01694, K01695, K01696, K01697, K01698, K01699, K01700, K01701, K01702, K01703, K01704, K01705, K01706, K01707, K01708, K01709, K01710, K01711, K01712, K01713, K01714, K01715, K01716, K01717, K01718, K01719, K01720, K01721, K01722, K01723, K01724, K01725, K01726, K01727, K01728, K01729, K01730, K01731, K01732, K01733, K01734, K01735, K01736, K0

No geral, o Cytoscape prioriza uma interface de usuário limpa e funcional que fornece acesso direto às ferramentas de visualização e análise de rede. Embora possa não ter uma única tela inicial, o *layout* e a organização são projetados para agilizar seu fluxo de trabalho para explorar e analisar dados de rede.

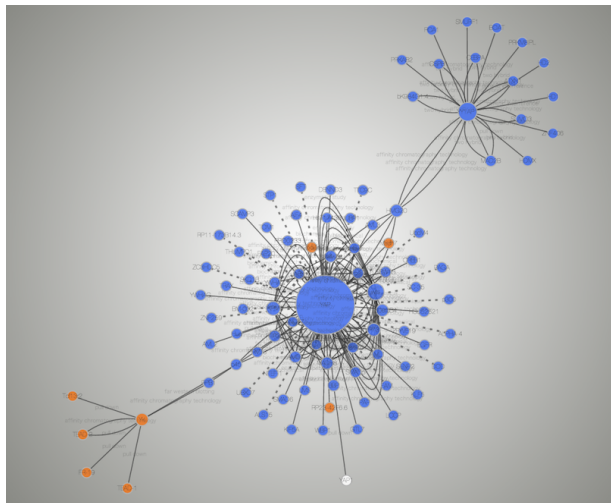
Funcionalidades

Cytoscape.js é uma biblioteca JavaScript de código aberto projetada especificamente para criar visualizações de rede interativas em navegadores da *web*. Ele foi projetado para ser integrado facilmente em aplicações *web* e com ela pode-se criar visualizações de grafos interativas e personalizáveis. O Cytoscape.js também oferece muitas opções de estilo e interação para personalizar a aparência e o comportamento da visualização. Essa ferramenta é uma opção popular para visualização de grafos em aplicações web, pois é fácil de usar e permite a criação de visualizações avançadas. Além disso, ele é compatível com vários formatos de dados de grafos, incluindo JSON e GEXF, e é compatível com dispositivos móveis.

O Cytoscape.js é baseado em uma biblioteca de visualização de grafos chamada D3.js, e conta com uma série de recursos avançados para análise de redes, como *layout* de grafos, análise de componentes e métricas de centralidade. Ele também permite personalizar da aparência da visualização de grafos através de estilos CSS e oferece suporte a animações e interações com o usuário. A biblioteca também oferece suporte a OpenCL, que é uma plataforma de computação de alto desempenho que permite acelerar o cálculo de métricas de rede em hardware de GPU. Isso permite que o Cytoscape.js execute análises em grandes conjuntos de dados de maneira mais rápida. Um exemplo interativo, pode ser acessado clicando aqui. Esse é um exemplo nativo hospedado pelo GitHub oficial do Cytoscape. Através da figura 3.5 é possível notar que a visualização é praticamente a mesma na versão *web* e na versão *desktop*.



(a) Visualização na aplicação desktop



(b) Visualização no Cytoscape.js

Figura 3.5: Visualização de grafos na aplicação *desktop* e no Cytoscape.js do mesmo grafo [3]

Outro diferencial da ferramenta é o Cytoscape *Automation*, essa ferramenta permite automatizar tarefas e processos. Particularmente útil quando é preciso realizar muitas tarefas repetidas de o Cytoscape for maneira em fluxos de trabalho mais complexos. Essa ferramenta de automação é baseada em *scripts* e pode ser usada para realizar ações como criar ou modificar redes, aplicar algoritmos de análise, modificar a aparência de redes e exportar imagens. Ela pode ser usada tanto por linhas de comando quanto por meio de uma interface gráfica. Além disso, o Cytoscape *Automation* é compatível com outras linguagens de programação, como Python, Java e R. O artigo *Cytoscape Automation: empowering workflow-based network analysis* [23], de 2019, explora o Cytoscape *automation* como um avanço significativo para os usuários do Cytoscape. Ao permitir a análise de rede baseada em fluxo de trabalho, ele agiliza o processo de pesquisa, melhora a reprodutibilidade e capacita os pesquisadores a aproveitar as funcionalidades do Cytoscape de maneira mais profunda.

Desenvolvimento de plugins e scripts

Apesar de ser uma ferramenta completa, ainda existe a possibilidade de desenvolver *plugins* para sanar alguma necessidade não resolvida com as ferramentas nativas do Cytoscape. Esses *plugins* permitem que sejam adicionadas novas funcionalidades, algoritmos de análise de dados ou estilos de visualização adaptados especificamente às suas necessidades de pesquisa e investigação. Os *plugins* podem ser desenvolvidos em Java ou JavaScript,

duas linguagens poderosas e robustas para criação de algoritmos. Além disso, a comunidade do Cytoscape é ativa, podendo fornecer *plugins* prontos e que possam ser usados sem a necessidade de desenvolver um novo. Caso não seja suficiente os *plugins* disponibilizados pela comunidade, pode ser que os mesmos sirvam como base para o desenvolvimento específico de um *plugin* para atender a necessidade do projeto.

Formatos aceitos para importação e exportação

Cytoscape suporta uma variedade de formatos de arquivo de rede, permitindo importar dados de várias fontes. Dentre as opções, os arquivos padrão para redes são: SIF, PSI-MI, GraphML, .xlsx, CSV, XGMML e GEXF. A ferramenta também pode importar dados de rede de arquivos de texto simples onde nós e bordas são separados por delimitadores como tabulações ou vírgulas.

Para exportação, semelhante à importação, o Cytoscape oferece uma ampla variedade de opções de exportação para os dados da rede. Os formatos de rede padrão: SIF, PSI-MI, GraphML, XGMML e GEXF. Adicionalmente é possível exportar para formatos de imagem: PNG, JPEG, BMP e TIFF. Além desse, também é possível exportar para formatos vetoriais como o SVG. Por fim, é possível exportar para o formato de arquivo da própria ferramenta, o .cys, que permite reabrir a rede e sua configuração visual em uma sessão futura do Cytoscape.

3.3.2 NetworkX

Como definido no próprio site da ferramenta [24], *NetworkX é um pacote Python para criação, manipulação e estudo da estrutura, dinâmica e funções de redes complexas*. A biblioteca NetworkX é baseada em uma estrutura de grafos. Ela permite que os usuários criem grafos a partir de dados e os manipulem usando diferentes algoritmos e funções. Vale ressaltar que os dados podem ser "qualquer coisa", como imagens, textos, XML, desde que possa ter um identificador único. O Networkx tem uma série de funcionalidades muito boas para se trabalhar com SNA e conta com vários algoritmos para aplicar nos dados. Como a ferramenta é um pacote do Python, é possível utilizar várias bibliotecas, funções e recursos da linguagem em conjunto com o Networkx. A ferramenta possui uma documentação completa e muitos usuários, sendo um ponto de apoio para resolver problemas no decorrer do uso e da configuração. Ela é compatível com Windows, MacOS e Linux, tornando-o uma opção flexível e acessível para trabalhar com análise de redes e grafos em diversas plataformas. A licença para uso do Networkx é a *BSD 3-Clause*.

Interface

O NetworkX não tem uma interface nativa para se trabalhar, como é possível verificar na figura 3.6. Ela tanto pode ser incorporada a alguma interface, quanto pode ter a sua própria através do desenvolvimento. Com a ferramenta é possível construir um grafo de interações do zero usando os comandos do NetworkX em consonância com alguma biblioteca para criar as imagens das redes. Nos exemplos abordados no presente trabalho, a biblioteca Python usada para geração das imagens dos grafos foi a *matplotlib*, indicada pelo próprio tutorial da ferramenta.

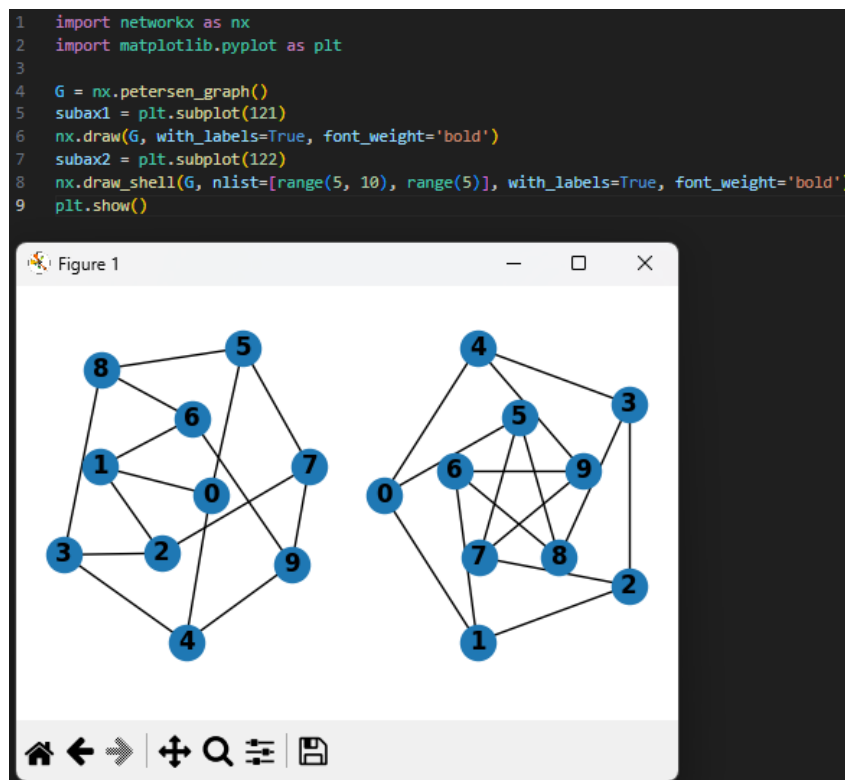


Figura 3.6: Esboço inicial de um grafo simples usando o NetworkX.

Funcionalidades

NetworkX se destaca entre as ferramentas de análise de rede por sua facilidade de uso. Sua API é intuitiva e a documentação é clara, o que a torna mais acessível. A ferramenta possibilita os usuários na criação, manipulação e análise de redes com muita facilidade, o que a torna uma ferramenta fácil de ser incorporada em outros projetos, como é o caso do ecossistema SmartUnB.ECOS. Essa facilidade de uso agiliza o processo de aprendizagem e permite que os usuários se concentrem nos *insights* que desejam extrair de seus dados, e não no domínio da interface do software.

Mesmo sem uma interface de usuário, o Networkx trabalha muito bem com os dados disponibilizados a ele. Como dito na introdução da ferramenta e no próprio site, os nós usados no NetworkX podem ser "qualquer coisa". Porém, cabe uma explicação sobre essa peculiaridade da ferramenta. NetworkX se concentra principalmente nos relacionamentos estruturais entre entidades em uma rede. Para isso, é necessário que essas entidades (ou nós), sejam únicas e identificáveis na rede. Por esse motivo, os nós precisam ter um identificador único, um *hash*. Com os nós identificados, a ferramenta se concentrará nos relacionamentos entre dois ou mais nós, analisando as ligações e os relacionamentos deles. Mas como o NetworkX pode trabalhar com "qualquer coisa"? Embora o NetworkX não incorpore diretamente dados complexos como imagens ou XML nos nós, ele oferece duas estratégias para trabalhar com esses dados. A primeira, são as classes personalizadas. Com elas os usuários podem definir classes personalizadas para encapsular tipos de dados complexos, como texto, URLs de imagem ou referências a objetos XML externos. Essas classes atuam como “contêineres” para os dados reais, e o nó no NetworkX simplesmente contém uma referência à instância da classe personalizada. A segunda estratégia para se trabalhar com dados complexos são os atributos do nó. Com eles, NetworkX permite designar atributos a nós individuais. Esses atributos podem ser de qualquer tipo de dados, incluindo dicionários contendo texto, URLs de imagens ou até mesmo referências a fontes de dados externas. Esta abordagem separa novamente o próprio nó (que é um identificador) dos dados associados armazenados no atributo.

Desenvolvimento de plugins e scripts

Por se tratar de uma biblioteca Python, o desenvolvimento de *plugins* e scripts para o NetworkX devem ser desenvolvidos em Python. Isso é uma grande vantagem pela extrema facilidade de uso da linguagem. Além disso, o ecossistema Python é extenso, o que permite fazer integrações das funcionalidades do Networkx com outras ferramentas, sistemas e até mesmo outras bibliotecas de análise de redes. A linguagem de programação pode ser um forte aliado da ferramenta por possibilitar a cobertura de um dos principais pontos negativos do NetworkX, que são os recursos de visualização limitados. Embora o NetworkX ofereça funcionalidades básicas de visualização de rede, ele pode não ser tão poderoso ou fácil de usar para visualização avançada.

Formatos aceitos para importação e exportação de dados

Os formatos que são possíveis de importar para o Networkx são: Edge list, Node list (adjacency list), GraphML (.graphml), Pajek (.paj), GEXF (.gexf), Matrix Market, Network Text format, sendo os três últimos os formatos menos usados na ferramenta. Formatos não suportados pela ferramenta podem ser lidos usando funções Python.

Já para exportar dados, os formatos são: Edge list, Node list (adjacency list), GraphML (.graphml), GEXF (.gexf), Pajek (.paj), NetworkX pickle format (.pickle), JSON, Matrix Market, DOT language (.dot). É possível usar funções Python para exportar os dados para Formatos não suportados pela ferramenta.

3.3.3 Socnetv

SocNetV é um software projetado especificamente para atender às necessidades de análise de redes sociais. Ele fornece uma interface amigável que permite aos usuários criar, manipular e analisar redes sociais. Esta abordagem centrada no usuário torna o SocNetV particularmente atraente para usuários que podem não possuir ampla experiência em programação. Ao contrário das ferramentas que exigem scripts complexos ou conhecimento de codificação, o SocNetV permite aos usuários construir redes e realizar análises básicas por meio de menus intuitivos, o que agiliza a curva de aprendizagem e permite que os investigadores se concentrem nos fenômenos de rede, em vez de se preocupar com detalhes técnicos. Um dos pontos fortes do SocNetV são suas capacidades de visualização. Ela oferece ferramentas de visualização integradas que permitem aos usuários gerar representações de rede claras e informativas. Essas visualizações podem ser personalizadas usando diferentes layouts de nós e esquemas de cores, permitindo que os usuários adaptem os recursos visuais para destacar aspectos específicos da rede. O ponto chave dessa ferramenta é principalmente esse, as funcionalidades para visualização estão integradas e é uma ferramenta muito amigável para usuários inexperientes com programação. O site oficial da ferramenta pode ser acessado em [4]

Interface

O SocNetV é uma ferramenta com uma interface muito amigável (Figura 3.7), contendo semelhanças e botões muito parecidos com outras interfaces conhecidas no mercado de softwares, como editores de imagem, texto ou vídeo. Os botões para criar e remover nós, importar e exportar arquivos, aplicar *zoom* e filtros e o painel de controle do grafo estão facilmente rastreáveis na tela e com imagens que representam bem suas funções. O painel de controle do grafo da mesma maneira, possui descrições e opções de edição do grafo simples e diretas. Pelo fato de a imagem do grafo ficar centralizada na tela, o foco no conteúdo é garantido, mas sem deixar de lado as funcionalidades e a facilidade de usá-las.

Para mostrar mais a ferramenta e a facilidade de configuração dos nós e arestas e como o grafo pode ser personalizado com facilidade, vamos explorar um pouco das configurações gerais da ferramenta, que podem ser vistas na figura 3.9. O SocNetv oferece a possibilidade de personalizar os nós do grafo, inserindo imagens ou outras formas desejadas pelo usuário

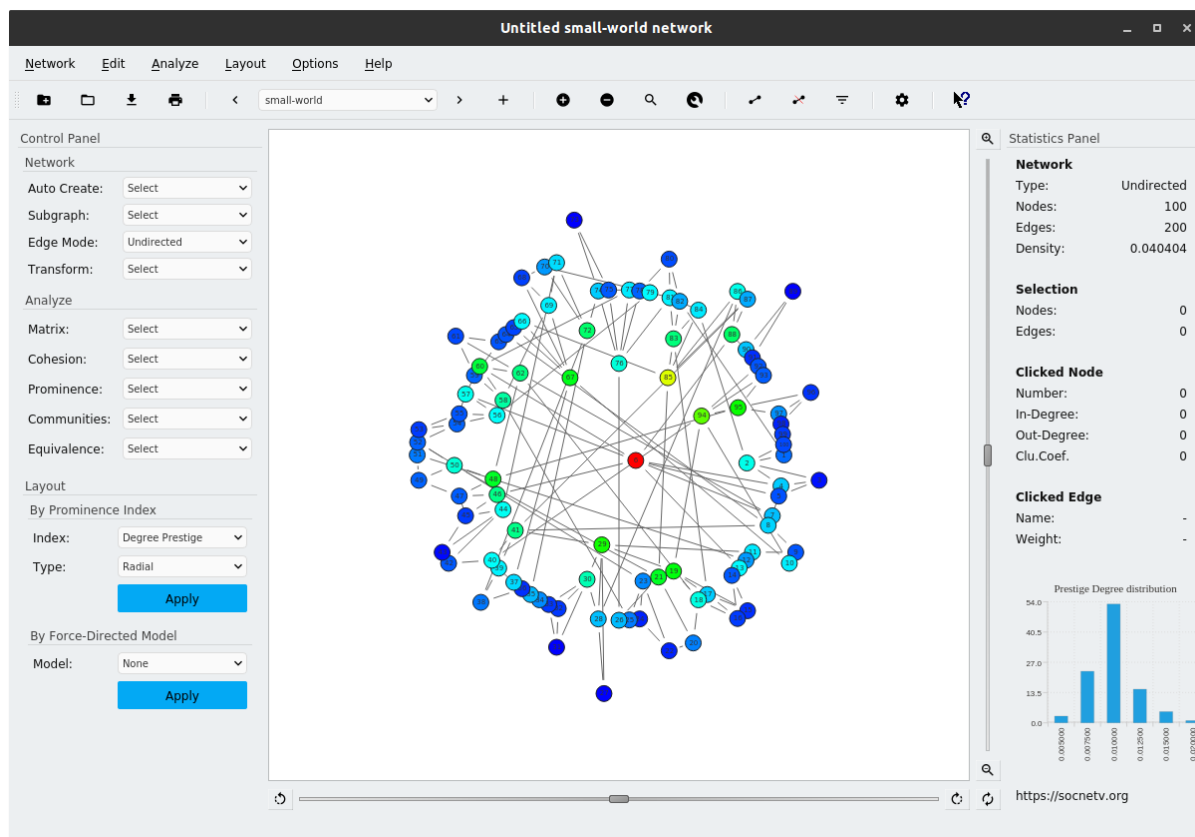


Figura 3.7: Interface *user friendly* do SocNetv [4].

e que ajudem na compreensão das relações estabelecidas nele. A Figura 3.8 mostra um exemplo onde os nós são pessoas e empresas, clareando bastante o entendimento das relações.

A interface do SocNetv foi produzida priorizando a alta qualidade usando o *Material Design*, garantindo consistência entre sistemas operacionais. Uma aparência uniforme em Windows, Mac e Linux elimina elementos de UI específicos da plataforma, reduzindo a confusão e tornando o software mais intuitivo para usuários familiarizados com qualquer um dos principais sistemas operacionais. Essa consistência promove uma experiência adequada, independentemente do sistema operacional.

Funcionalidades

O SocNetv possui um *Web Crawler*, que foi recém atualizado trazendo melhorias significativas para essa ferramenta. O *Web Crawler* atua como uma ferramenta capaz para extrair automaticamente dados de sites e construir uma representação correspondente. Ele identifica conexões (arestas) entre páginas da web (nós) com base em *hyperlinks*. Seguindo *links*: o rastreador inicia a partir de um URL inicial fornecido pelo usuário (a

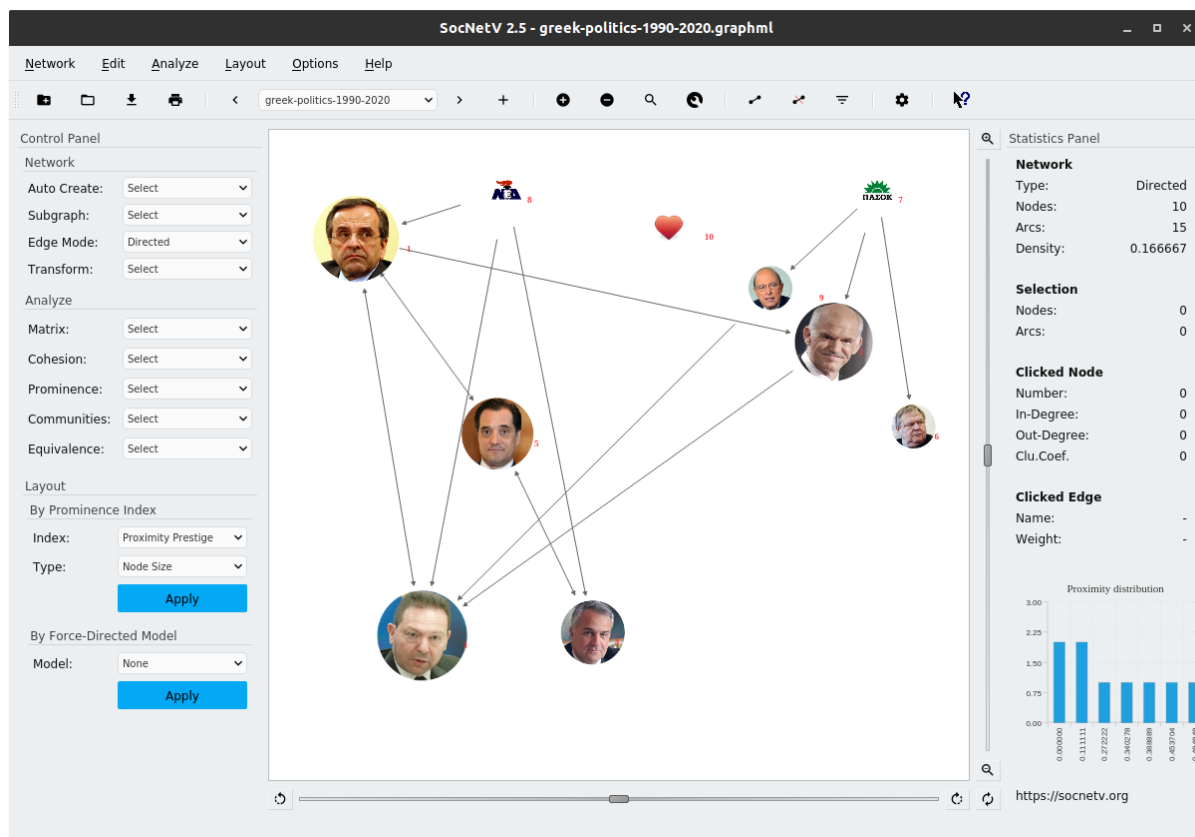


Figura 3.8: Construção de um grafo personalizado no SocNetv [4].

página inicial) e segue os *hyperlinks* que encontra nessa página. Ele pode percorrer vários níveis na estrutura do site, dependendo das configurações do usuário. À medida que o rastreador explora o site, ele constrói uma estrutura de rede dentro do SocNetV. Cada página da web se torna um nó na rede e os *hyperlinks* entre as páginas são traduzidos em arestas que conectam os nós correspondentes.

Para tanto, o usuário pode configurar como deseja que o *Web Crawler* atue, podendo controlar quais *links* o rastreador segue, optar por incluir *links* entre páginas pai e filho na estrutura do site e especificar padrões ou palavras-chave para excluir *links* normalmente associados a plataformas de redes sociais. Isso ajuda a focar no conteúdo do site e evita conexões irrelevantes.

Desenvolvimento de plugins e scripts

O Socnetv é uma ferramenta poderosa, mas uma de suas limitações é o desenvolvimento de *plugins*, pois ele não oferece um ambiente tradicional de desenvolvimento de *plugins*. No entanto, existe uma abordagem para incorporar algum nível de automação usando as macros, que são a única funcionalidade de *script* integrada no SocNetV. Eles permitem que

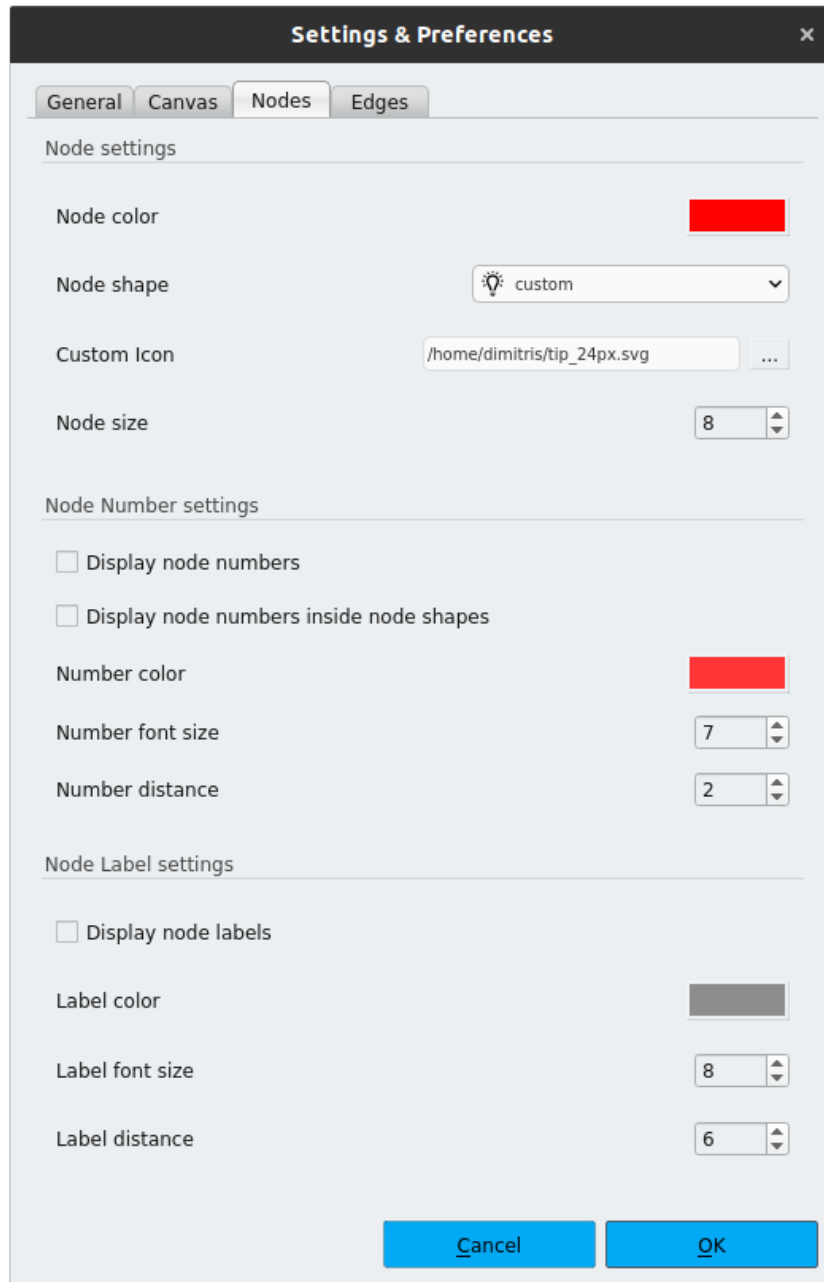


Figura 3.9: Configurações gerais do SocNetv [4].

se registre uma sequência de ações executadas no software. Depois de gravada, você pode reproduzir a macro para automatizar tarefas repetitivas. Com macros ainda é possível aplicar o mesmo layout de rede a múltiplas redes, padronizar a aparência visual de nós e arestas em diferentes redes e automatizar transformações básicas de dados, como filtrar nós com base em critérios específicos. Porém, ainda assim, macros são boas apenas com ações simples e sequenciais, ou seja, aplicando lógicas simples, não sendo adequadas para tomada de decisão dentro do processo de automação. A linguagem da ferramenta é o

Python, mas, o Socnetv não oferece uma forma de integrar os *plugins* desenvolvidos na ferramenta, sendo necessário fazer qualquer processamento fora da aplicação.

Formatos aceitos para importação e exportação

SocNetV prioriza GraphML (Graph Markup Language) como seu formato de arquivo principal para salvar e carregar dados de rede. Apesar de favorecer o GraphML, o SocNetV oferece flexibilidade ao permitir que os usuários importem dados de vários outros formatos de arquivo de rede, como GML (Graph Modeling Language), Pajek, DOT e matriz de Adjacência e lista de arestas. Matriz de adjacência e lista de arestas são formatos de arquivo de texto.

Apesar de o SocNetV importar vários formatos, a exportação de dados está atualmente limitada ao GraphML e alguns formatos de imagem. Os formatos de imagem são PNG, JPG, BMP, TIFF. Também existe a possibilidade de exportar para PDF. Mesmo não sendo formatos muito interessantes para o contexto do ecossistema, pode ter sua utilidade em outros aspectos. Apesar da dificuldade de exportar apenas para GraphML, existe a possibilidade de converter o arquivo para um formato desejado, contornando assim essa questão na ferramenta.

3.3.4 Gephi

Gephi é uma ferramenta de visualização de dados de código aberto, desenvolvida em Java e é muito utilizada na análise de redes sociais. Ele permite aos usuários carregar, visualizar e manipular dados de rede para explorar padrões e relacionamentos. Além disso, fornece uma série de recursos para visualizar e analisar dados, além da integração com outras ferramentas de análise de dados. Uma das principais vantagens do Gephi é a sua capacidade de lidar com grandes conjuntos de dados, possibilitando a visualização de redes com milhares de nós e arestas. Ele também suporta diferentes formatos de dados de entrada e consegue exportar seus dados para vários formatos interessantes para se trabalhar com redes. Além disso, Gephi oferece uma plataforma de programação aberta para o desenvolvimento de *plugins* personalizados para automatizar tarefas e adicionar alguma funcionalidade. É uma ferramenta muito acessível para uso, devido a ter versões instaláveis para Windows, MacOS e Linux. Para fomentar o uso científico da ferramenta, Gephi tem um artigo publicado na *Association for the Advancement of Artificial Intelligence*, IAAA [25]. O artigo discute os recursos da ferramenta, como o uso do mecanismo de renderização 3D para exibir grandes redes. O artigo também detalha a capacidade do Gephi de espacializar, filtrar, navegar, manipular e agrupar dados de rede.

Por fim, destaca os recursos de visualização dinâmica de rede. Essas informações podem ser encontradas no site oficial da ferramenta [5].

Interface

A interface do Gephi oferece um espaço de trabalho (*workspaces*) flexível para análise e visualização de redes, pois cada projeto em que se trabalha é um *workspace* separado, para gerenciar diferentes análises de rede de forma independente. Ou seja, é possível abrir visualizações independentes ao mesmo tempo, com suas próprias configurações e customizações. A interface é dividida em três painéis principais, que ficam como abas na parte superior da ferramenta. As abas são:

Overview (Visão geral): Este é o *workspace* principal onde se pode manipular o grafo da rede, alterando cores dos nós e arestas e aplicando algoritmos de layout. Vale ressaltar que é possível fazer algumas customizações também em outros menus. A figura 3.10 mostra a interface desse *workspace*.

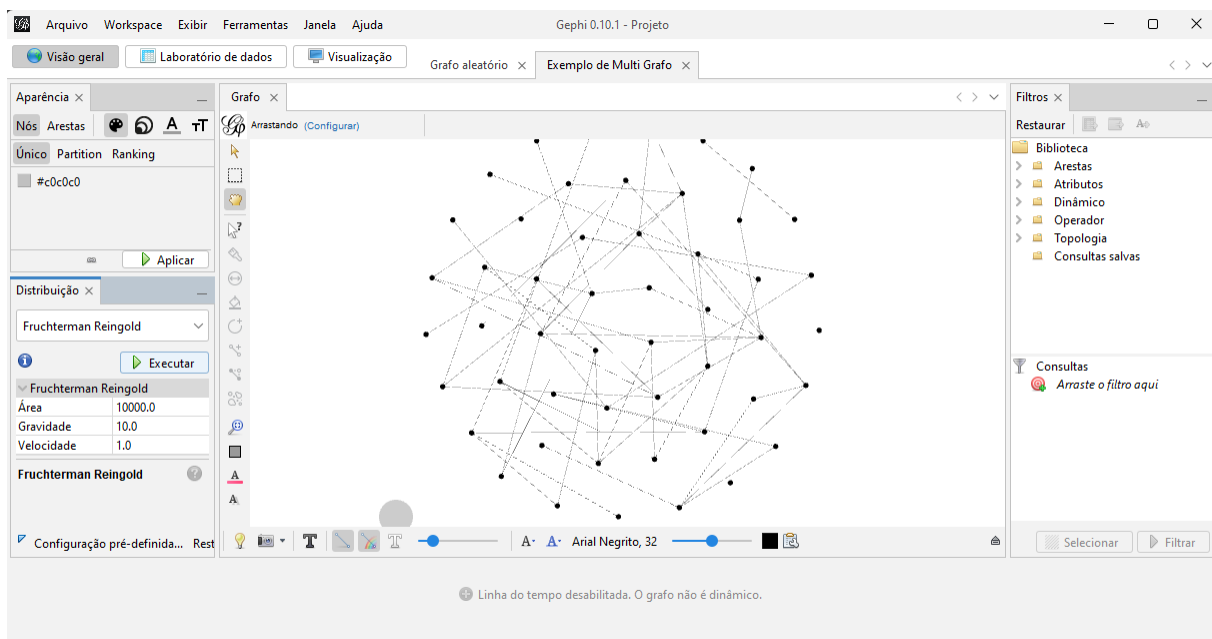


Figura 3.10: Guia de Visão geral no Gephi [5].

Data Laboratory (Laboratório de Dados): É um *workspace* que oferece uma interface semelhante ao Excel para os usuários manipularem colunas de dados, como pode ser visto na figura 3.11. Nessa área, estão presentes ajustes como pesquisar e substituir, manipulação de colunas, edição em lote (usando critérios pré configurados). O Laboratório de Dados atua como um *hub* central para gerenciar os dados associados a rede.

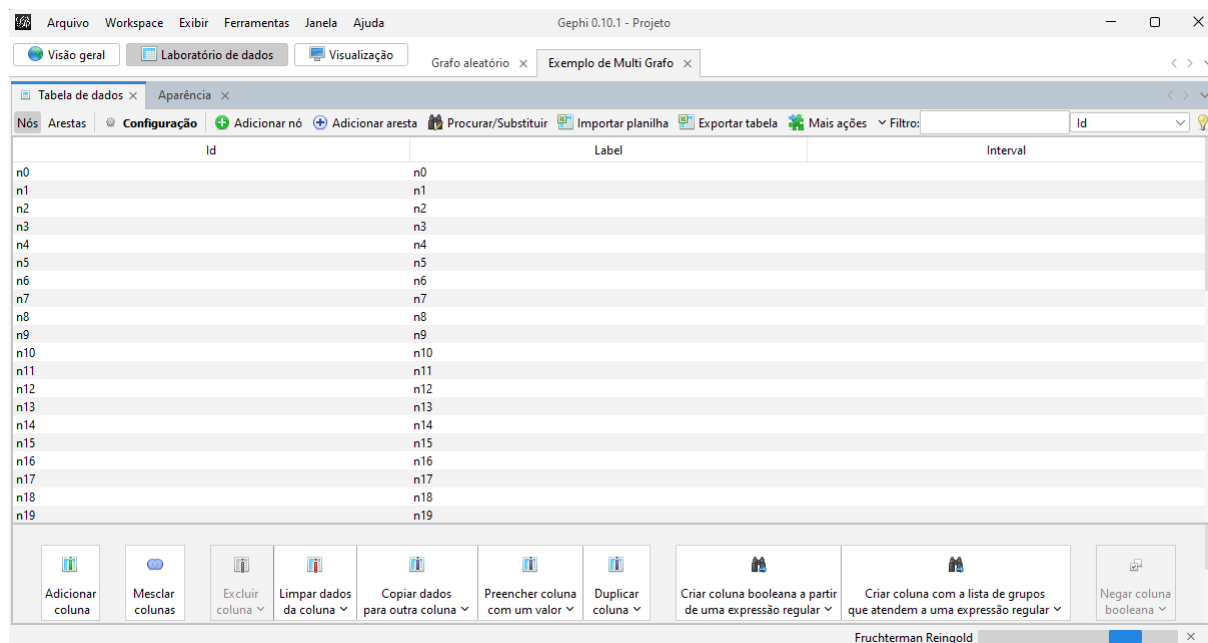


Figura 3.11: Laboratório de dados no Gephi [5].

Preview (Pré-visualização): Esta seção permite ajustar a apresentação visual da rede antes da exportação. Aqui é possível editar alguns aspectos da visualização da rede, como a cor, descrição, borda, entre outros, vide figura 3.12.

A interface conta também com as opções padrão para aplicar análise de métricas, filtragem (Figura 3.13) e personalizar profundamente os nós e arestas da rede. Nos próprios menus superiores é possível acessar os *plugins* disponíveis para a ferramenta. Um dos *plugins* que mais fazem sentido para o projeto seriam o JSONExporter, para exportar os dados para o formato JSON. A medida que opções são escolhidas nos menus superiores, os painéis correspondentes vão sendo abertos nas laterais, possibilitando customizar e alterar opções simultaneamente, sem necessidade de mudança de janelas (para o *workspace* atual). No menu janela, é possível gerar estatísticas da rede de maneira automática, apenas selecionando o que se deseja saber sobre a rede e pressionar o botão executar, então as métricas são geradas em uma janela separada, referente ao estado atual da rede. É muito fácil de ser usada, sendo necessário apenas um clique para gerar gráficos sobre a rede. É possível também juntar as janelas das ferramentas apenas arrastando e soltando, mostrando a versátil área de trabalho do Gephi. O mesmo acontece para aplicação de filtros por exemplo. Nesse menu existem várias opções já pré configuradas, onde o usuário precisa apenas selecionar qual deseja usar e arrastar para a área de filtros.

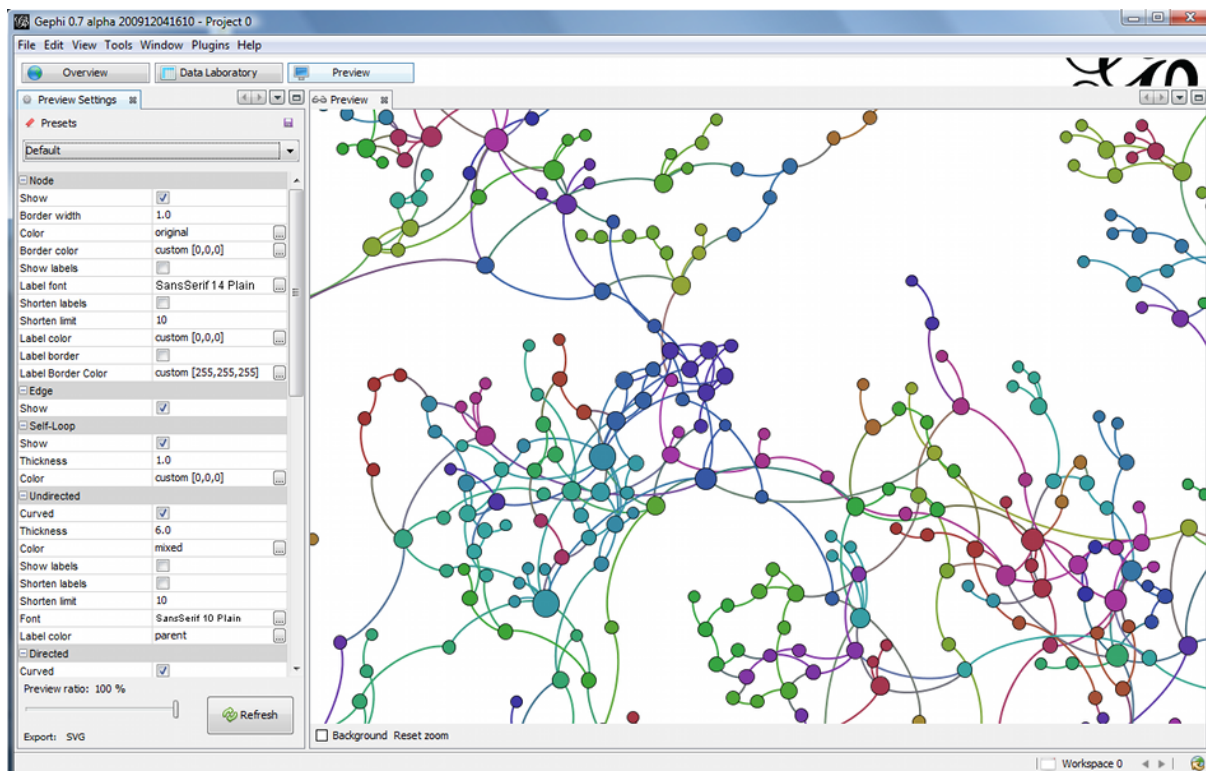


Figura 3.12: Área de pré visualização no Gephi [5].

Funcionalidades

Um dos pontos mais fortes do Gephi é excelente na criação de representações de rede visualmente informativas e personalizáveis. Como dito anteriormente, o Gephi trabalha muito bem com redes complexas e grandes conjuntos de dados. É nesse momento que a visualização da ferramenta entra em ação. O poder de personalização (Figura 3.14) e de visualização das informações é muito amplo, tendo como destaque a visualização 3D dos dados da rede. O site da ferramenta apresenta um vídeo introduzindo as principais características do Gephi e ele pode ser visualizado clicando aqui. Um dos elementos apresentados é o mecanismo de visualização de grafos, que funciona através de um mecanismo OpenGL *ad-hoc*. Com esse mecanismo, a ferramenta é capaz de rapidamente carregar a visualização de redes de até 100.000 nós e 1.000.000 de arestas e iterar através da visualização usando filtragem dinâmica.

Outro diferencial do Gephi é a capacidade visualizar redes que evoluem ao longo do tempo, também conhecidas como redes dinâmicas ou redes temporais, vide figura 3.15. O Gephi possui uma funcionalidade para visualizar como essa rede evoluiu ao longo do tempo, manipulando a linha do tempo incorporada para visualizar como a rede foi modificada com o passar do tempo.

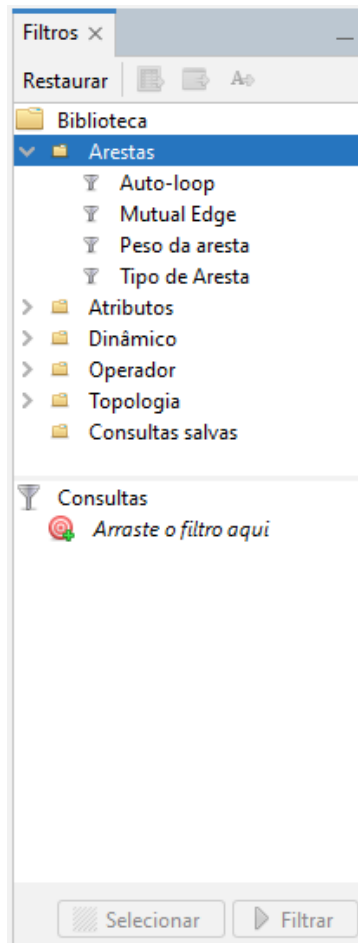


Figura 3.13: Fácil aplicação dos filtros na ferramenta Gephi [5].

Para trabalhar com essas redes dinâmicas, existem duas maneiras:

- Série de redes estáticas: você pode dividir os dados dinâmicos da rede em uma série de instantâneos que representam a rede em diferentes momentos. Cada *snapshot* pode ser tratado como uma rede estática separada no Gephi. Com isso, é possível realizar a análise de rede social para comparar essas redes estáticas e identificar mudanças na estrutura da rede ao longo do tempo.
- GEXF com atributos dinâmicos: O formato de arquivo GEXF pode ser estendido para incluir atributos dinâmicos, possibilitando especificar o intervalo de tempo que um nó ou borda existe na rede. Assim, é possível usar o Gephi para analisar essas informações e permitir filtragem e visualização com base em períodos de tempo específicos.

Outra questão sobre a ferramenta que vale mencionar é que, semelhante ao Cytoscape, ele possui uma versão web, chamada Gephi Lite, a exemplo da figura 3.16. Ela tem os

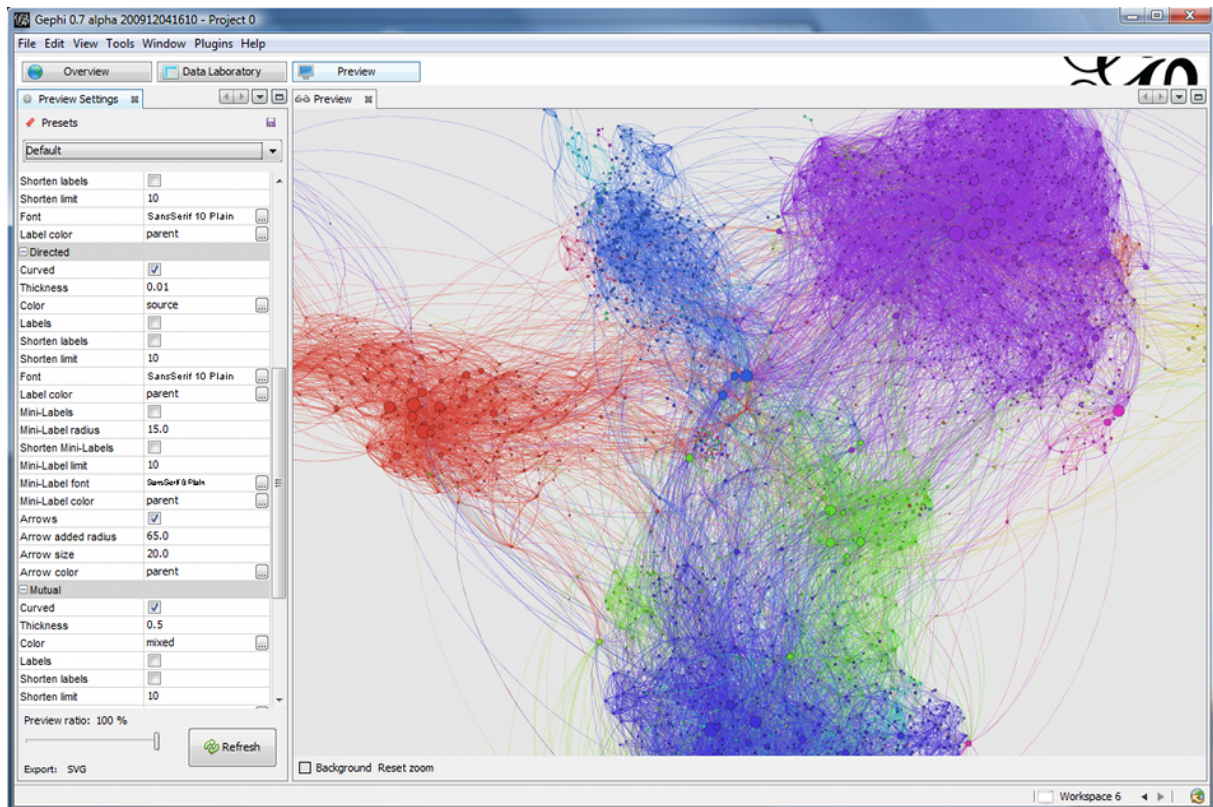


Figura 3.14: Personalização dos elementos da rede no Gephi [5].



Figura 3.15: Linha de tempo de uma rede no Gephi.

mesmos recursos básicos do Gephi, mas reduzido a um pacote com menos funcionalidades. Ele é semelhante ao Gephi e é compatível com ele, mas tem menos opções. Não é possível abrir grandes redes, mas é mais simples e ergonômico. Ainda conta com algoritmos de layout, personalização e filtragem da rede. Funcionalidades mais avançadas não estão presentes nessa versão, como detecção de comunidade, cálculos de métricas de rede e análises estatísticas. Essa versão pode ser acessada em [6]. Ela atende usuários que trabalham com redes menores e tem como objetivo oferecer uma experiência de usuário simplificada em comparação com o software completo. Um ponto de observação é que essa ferramenta aceita como importação apenas o formato GEXF e exporta para o formato de imagem.

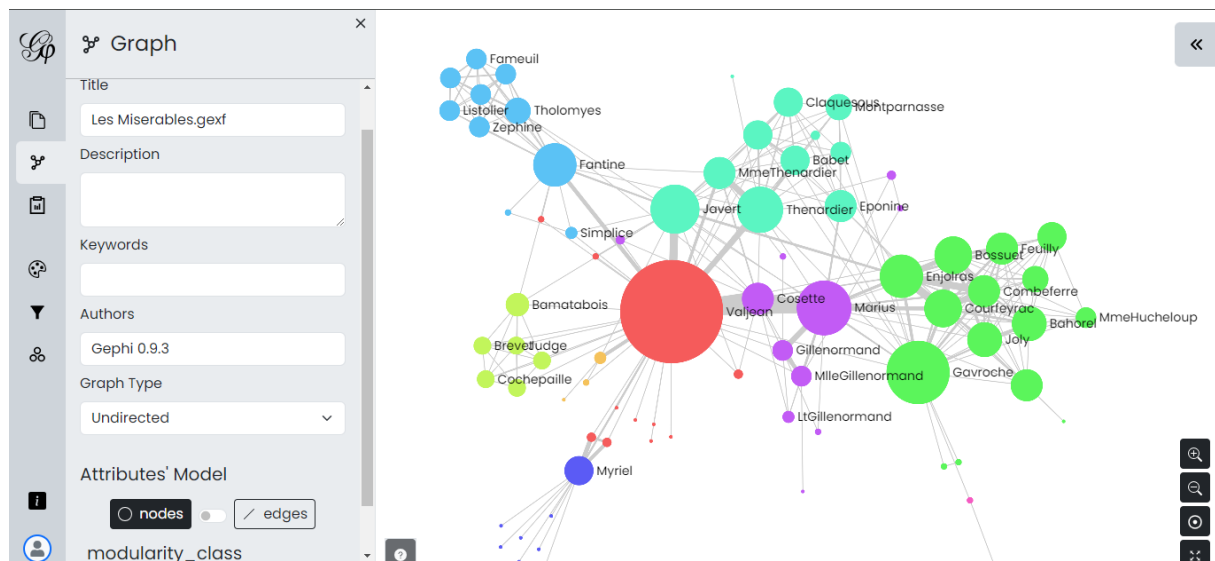


Figura 3.16: Exemplo de rede vista através do Gephi Lite [6].

Desenvolvimento de plugins e scripts

O Gephi foi desenvolvido principalmente em Java. Isso possibilita uma rápida integração com qualquer sistema operacional, inclusive, facilitando o desenvolvimento de *plugins* e *scripts*. O Gephi possibilita o desenvolvimento de *plugins* para adicionar novas funcionalidades específicas à ferramenta. Os *plugins* podem ser escritos em várias linguagens JVM, como Java, Scala ou Groovy e para facilitar esse processo, o Gephi fornece uma API de *plugins* e documentação para orientar os desenvolvedores na criação de *plugins*. Além disso, a ferramenta possui *plugins* existentes, que podem já contemplar a necessidade procurada com o desenvolvimento. Os *plugins* disponíveis podem ser encontrados em <https://gephi.org/plugins/> ou acessando diretamente da ferramenta, através do menu *Tools*. Na parte de *scripts*, o Gephi conta com as macros para automatizar rotinas e tarefas simples, como dito anteriormente.

Formatos aceitos para importação e exportação

O Gephi oferece várias opções de importação e exportação para facilitar o trabalho com dados de rede de diversas fontes. Para importação, seu principal formato é o GEXF (*Gephi Exchange Format*), que é o formato nativo do Gephi. Ele pode armazenar dados de rede, atributos de nós e bordas e informações de layout. Além desse formato, ainda é possível importar dados dos formatos CSV, GraphML, GML (Graph Modeling Language), GDF (Graph Description Format) e Pajek NET. O Gephi ainda realiza uma validação de dados básica durante a importação para identificar possíveis erros ou inconsistências em seu arquivo de dados de rede.

Para exportação de dados, o Gephi disponibiliza como opções os formatos GEXF, GDF, GraphML e formatos de imagem, sendo eles PNG e SVG.

3.3.5 Graph-tool

Graph-tool é uma biblioteca C++ projetada para manipulação e análise estatística de redes com foco em desempenho e escalabilidade. Essa ferramenta foi desenvolvida em C++, com algumas ligações com Python, mas as principais estruturas de dados e algoritmos são implementados em C++, garantindo uma performance, gerenciamento da memória e eficiência dos algoritmos usados. Apesar disso, a interação com as funcionalidades vem através do Python, que atua como uma ponte entre o núcleo C++ e o ambiente de *script* Python. As funcionalidades desenvolvidas em C++ são importadas como módulos Python. Segundo o próprio site da ferramenta, Isso confere um nível de desempenho comparável (tanto em uso de memória quanto em tempo de computação) ao de uma biblioteca C/C++ pura. Por isso, a ferramenta lida bem com redes complexas de forma eficiente, sem gargalos de desempenho.

Interface

Com relação a interface, a Graph-tool em si não possui uma interface gráfica, por se tratar primariamente de uma biblioteca do Python. Sendo assim, ela foi projetada para trabalhar com grafos em código, sendo a interface gráfica da preferência do usuário. Os resultados obtidos com o processamento de informações usando a ferramenta podem ser obtidos via linha de comando, através de *scripts* Python, sendo que para visualizar a rede é necessário usar bibliotecas gráficas, como o *matplotlib*. Existe a possibilidade de rodar a biblioteca em qualquer ambiente que aceite o Python, como notebooks Jupyter. Nesse caso porém, poderia não ser interessante para o projeto. A terceira opção para uma interface gráfica seria o desenvolvimento de um *frontend* que atendesse a necessidade e acionasse as opções desejadas. Na documentação oficial da ferramenta está presente um grafo gerado a partir de dados disponíveis no repositório da rede Netzschleuder. Com esse dados e usando os comandos de desenho de grafo, foi possível gerar um grafo da rede, que pode ser visto na Figura 3.17.

Uma coisa interessante de se citar é a possibilidade de usar a ferramenta para gerar grafos animados, atualizando o status da rede e registrando com a ajuda da função *graph_draw()*, que pode retornar e receber um objeto de janela GTK+ onde o desenho pode ser atualizado. Desta forma, animações simples podem ser feitas com muita facilidade, principalmente quando usadas em um *shell* interativo.

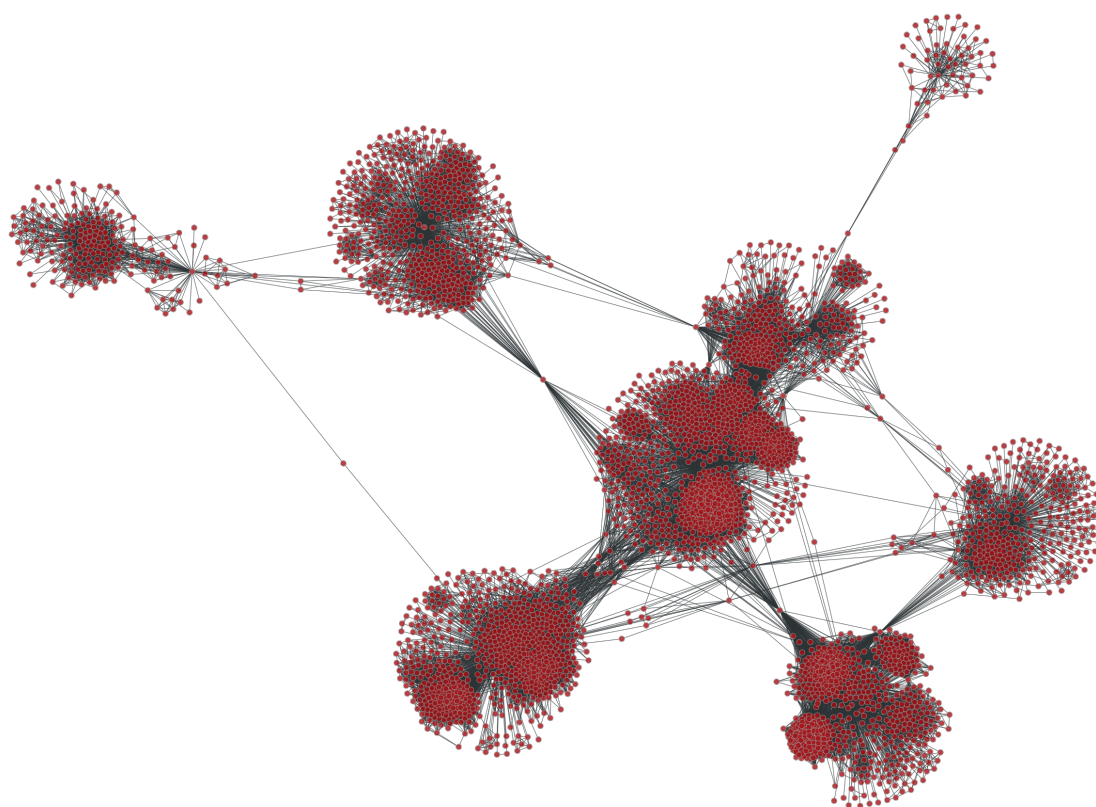


Figura 3.17: Grafo gerado com o Graph-tool usando dados da rede social Facebook [5].

Na criação dos grafos via linha de comando, cada nó e aresta (ou um conjunto deles) pode ser inserido de maneira já personalizada, inserindo cores, largura, formas, etc. Depois que a rede estiver pronta, é possível fazer todas as análises de métricas e filtros disponíveis nas ferramentas de SNA. Por fim, é possível desenvolver extensões em C++ para adicionar novas funcionalidades à ferramenta, mas esse tópico será melhor abordado adiante.

Funcionalidades

O ponto mais forte dessa ferramenta não é uma funcionalidade e sim sua robustez e desempenho. No site oficial [7] da ferramenta é possível encontrar uma comparação entre o Graph-tool e duas outras bibliotecas com implementações Python: Graph-tool, Igraph e NetworkX (Figura 3.18). A comparação avalia a rapidez com que cada biblioteca pode executar algoritmos específicos em um conjunto de dados específico e rodando em uma máquina com a mesma configuração. O objetivo é mostrar o desempenho do Graph-tool comparada com outras ferramentas feitas parcialmente (Igraph) e puramente (NetworkX) em Python. O resultado mostrou que em termos de velocidade, Graph-tool geralmente

é a mais rápida, especialmente ao usar vários núcleos. NetworkX é o mais lento porque está escrito em Python puro. Em termos de compilação e memória, Graph-tool leva mais tempo para compilar e requer mais memória, enquanto o Igraph usa menos memória para compilar e funciona com outras linguagens de programação.

| Algorithm | graph-tool (16 threads) | graph-tool (1 thread) | igraph | NetworkX |
|-----------------------------|-------------------------|-----------------------|---|--|
| Single-source shortest path | 0.0023 s | 0.0022 s | 0.0092 s | 0.25 s |
| Global clustering | 0.011 s | 0.025 s | 0.027 s | 7.94 s |
| PageRank | 0.0052 s | 0.022 s | 0.072 s | 1.54 s |
| K-core | 0.0033 s | 0.0036 s | 0.0098 s | 0.72 s |
| Minimum spanning tree | 0.0073 s | 0.0072 s | 0.026 s | 0.64 s |
| Betweenness | 102 s (~1.7 mins) | 331 s (~5.5 mins) | 198 s (vertex) + 439 s (edge) (~ 10.6 mins) | 10297 s (vertex) 13913 s (edge) (~6.7 hours) |

Figura 3.18: Comparação entre Graph-tool, NetworkX e Igraph. Imagem retirada do site da ferramenta [7].

A implementação de muitos dos algoritmos usados no Graph-tool foi feita com a ajuda do openMP, que é uma API que oferece suporte à programação de multiprocessamento de memória compartilhada multiplataforma em C, C++ e Fortran. Isso, segundo o site oficial, proporciona excelente desempenho em arquiteturas *multi-core*, sem degradá-lo em máquinas *single-core*, como pode ser visto na comparação entre as ferramentas acima. Além de ter um ótimo desempenho no uso de algoritmos, O Graph-tool conta com uma extensa gama de recursos, que vão além das funcionalidades para SNA comum a todas as ferramentas. A própria documentação da ferramenta menciona alguns exemplos como suporte para propriedades arbitrárias de vértices, arestas ou gráficos, filtragem eficiente "*on the fly*" de vértices e arestas, E/S de gráfico usando os formatos de arquivo GraphML, GML e DOT, estatísticas de grafos (histograma de grau/propriedade, correlações de vértices, distância média mais curta, etc.), medidas de centralidade, algoritmos topológicos padrão (isomorfismo, árvore geradora mínima, componentes conectados, árvore dominadora, fluxo máximo, etc.), geração de gráficos aleatórios com graus e correlações arbitrárias, detecção de módulos e comunidades por meio de inferência estatística, entre outros.

Desenvolvimento de plugins e scripts

O desenvolvimento de *plugins* se dá por meio da linguagem C++. O arquivo desenvolvido então é compilado e o arquivo de objeto é importado como um módulo Python. No Graph-tool não existe uma loja de *plugins* ou algo do tipo, todas as funcionalidades implementadas estão disponibilizadas com as bibliotecas importadas ao instalar a ferramenta.

Formatos aceitos para importação e exportação

De acordo com a documentação oficial da ferramenta, os formatos aceitos para importação são: GEXF, GraphML, pajek, lista de arestas e LGL. Para exportação, os formatos aceitos são: GEXF, GraphML, pajek e lista de arestas. Infelizmente não é possível importar e exportar arquivos diretamente de/para JSON ou CSV.

3.3.6 Igraph

Segundo o site oficial do Igraph [8], ela é uma coleção de ferramentas de análise de rede com ênfase em eficiência, portabilidade e facilidade de uso. Igraph pode ser usada em R, Python, Mathematica e C/C++. Sua portabilidade atua aqui, pois ela é uma ferramenta 4 em 1, podendo ser operada através das linguagens mencionadas anteriormente. Por ser um pacote, assim como outras ferramentas já mencionadas anteriormente, o Igraph também não possui interface gráfica, sendo acessada e acionada via linha de comando, vide figura 3.19. Possui estruturas de dados eficientes, otimizadas para representação de rede, o que a torna adequada para lidar com grandes redes, o que ajuda bastante no desempenho da ferramenta quando lida com grandes conjuntos de dados. Assim como as demais ferramentas abordadas, Igraph possui uma coleção abrangente de algoritmos para tarefas de análise de rede e de algoritmos de layout para o momento de visualização. Além disso, possui a capacidade de associar diversos tipos de dados a nós e arestas, assim, permite uma melhor compreensão dos elementos da rede e seus relacionamentos (Figura 3.20). A ferramenta possibilita a integração de *plugins* desenvolvidos pelo usuário para complementar suas funcionalidades ou adicionar algo específico a ela, que faça sentido para o usuário. Além disso, O Igraph importa e exporta para formatos conhecidos ao lidar com redes, entre eles GEXF e GraphML.

Interface

Assim como outras ferramentas que são bibliotecas de linguagens, o Igraph, da mesma forma, não possui uma interface de usuário pré programada. Em todas as diferentes linguagens em que pode ser executado, nenhuma delas conta com interface gráfica, ficando a cargo do usuário criar um *frontend* que atenda as suas necessidades. A ferramenta faz toda a parte de gerar a visualização através de suas funções para SNA, combinadas com bibliotecas gráficas, como, no caso do Python, a *matplotlib*.

Caso um *frontend* não seja necessário, pode-se criar uma automação através de *scripts* para geração das visualizações necessárias, combinando dados vindos de uma API por exemplo, ou gerados de outro sistema. Esses dados podem ser usados pela ferramenta nos formatos suportados para gerar os artigos solicitados via *script*. O Igraph pode também

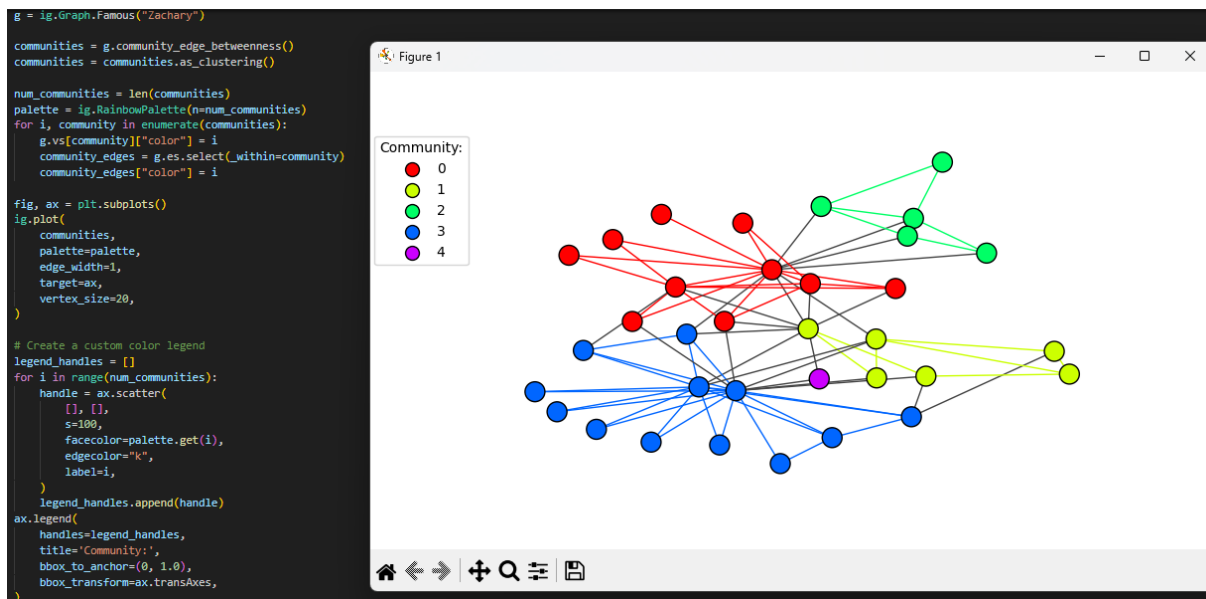


Figura 3.19: Exemplo de visualização de comunidade no Igraph.

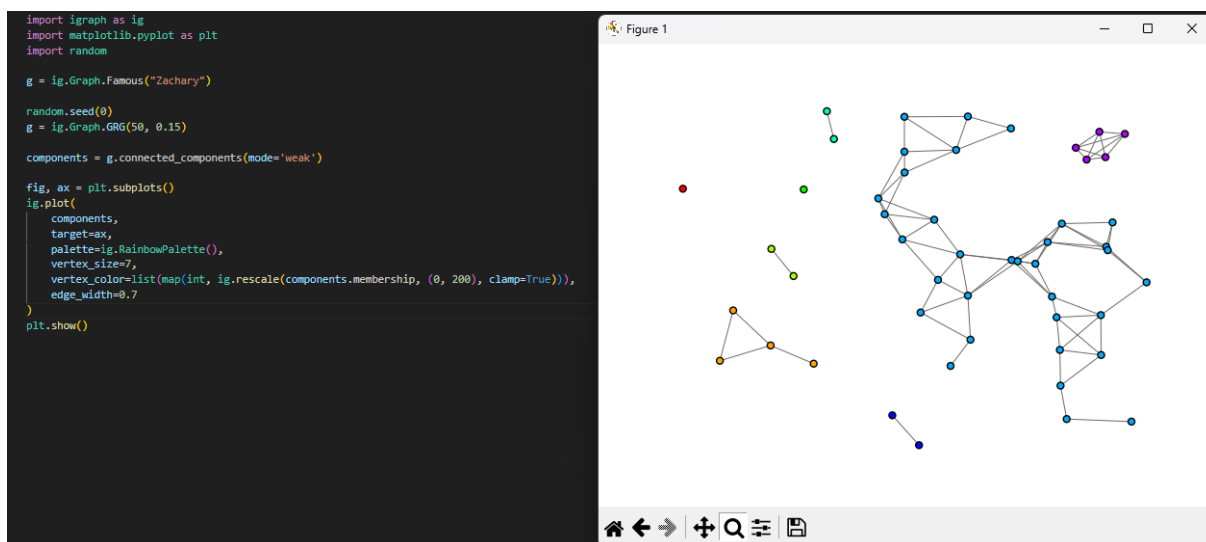


Figura 3.20: Exemplo de visualização de componentes conectados no Igraph [8].

ser integrado com outras ferramentas de SNA, como o NetworkX, para combinar algoritmos de layout ou aplicar outras funções antes de gerar a visualização. O Igraph tem uma documentação muito completa e que auxilia muito no processo de geração de visualizações de conceitos específicos, como *clusters*, identificação de ligações entre componentes da rede, entre outros. Vale ressaltar que os grafos gerados são uma representação bidimensional. Vários layouts podem ser aplicados em um grafo, e, normalmente, preservam ou destacam propriedades distintas. Na documentação oficial da ferramenta, existem bons exemplos de como e quais algoritmos de layout aplicar para obter resultados específicos,

como por exemplo, para gerar árvores existem algoritmos de layout indicados para isso, como layout *Reingold-Tilford* ou layout *Sugiyama*.

Funcionalidades

Em seu site oficial existe uma análise de como o Igraph se diferencia de seus concorrentes em alguns aspectos, trazendo seus pontos fortes frente a ferramentas como NetworkX e Graph-tool, que também são ferramentas no estilo biblioteca de linguagem. Sendo assim, primeiramente é abordado um ponto que torna o Igraph único: ele suporta múltiplas linguagens de programação, a se dizer C, Python, R e Mathematica. Isso o torna um grande aliado quando o assunto é ser acessível em diversos ambientes e para diversos propósitos.

No quesito desempenho, de forma similar ao Graph-tool, que tem um ótimo desempenho devido seu *core* ter sido desenvolvido em C++, o Igraph pode ter um desempenho tão bom quanto ele, pois sua biblioteca principal é escrita em C. Na análise, é levado em conta que a compilação do Igraph pode ser muito mais rápida e que ele é suportado em mais arquiteturas (hardware e software), pois a sua versão compilada exige menos (ou nenhuma) configuração e ajustes específicos dependendo do sistema onde será utilizado. Esse é um forte aliado do Igraph para uso em diferentes sistemas com arquiteturas variadas.

Tratando da parte de visualização em bibliotecas, o Igraph possui uma integração muito bem desenvolvida com o *matplotlib*, permitindo criar imagens estáticas de alta qualidade das redes, com grande capacidade de personalização. O diferencial do Igraph nesse sentido é seu suporte experimental (Está em fase experimental até o momento da produção desse texto) ao *plotly*, que é uma biblioteca para criação de visualizações interativas. Com ela, é possível criar animações dos dados da rede, mostrando como ela evolui com o tempo. Isso também se estende aos notebooks Jupyter, que podem usar os widgets para gerar elementos interativos, permitindo ajustar parâmetros ou filtrar os dados da rede diretamente na visualização. As outras ferramentas tem o suporte apenas a imagens estáticas.

Além disso, segundo o artigo de Csardi e Nepusz (2006) [9], que introduz a ferramenta, o Igraph foi desenvolvido com o intuito de fornecer um software capaz de lidar com grafos grandes (com milhões de vértices e arestas) de forma eficiente e com a possibilidade de ser usado de forma interativa ou não interativa. Porém, o ponto que se levanta no artigo é que quando se lida com grafos grandes, as interfaces não interativas podem ser mais adequadas, pois existem redes tão grandes que se levaria muito tempo em cálculos e métricas e isso é um dos motivos para se considerar interfaces não interativas, frente a tendencia das interfaces interativas.

Outro ponto interessante abordado no artigo da ferramenta, é que a arquitetura do Igraph é dividida em três camadas, que são conectadas através de interfaces. Porém, cada camada pode ser substituída por uma implementação alternativa sem alterar os outros componentes, o que aliado ao código *open source*, abre muitas possibilidades. A camada mais baixa contém as operações básicas e é implementada em C. A segunda camada contém quase todas as funções de análise de rede e também foram implementadas em C. A terceira camada contém as interfaces de nível superior.

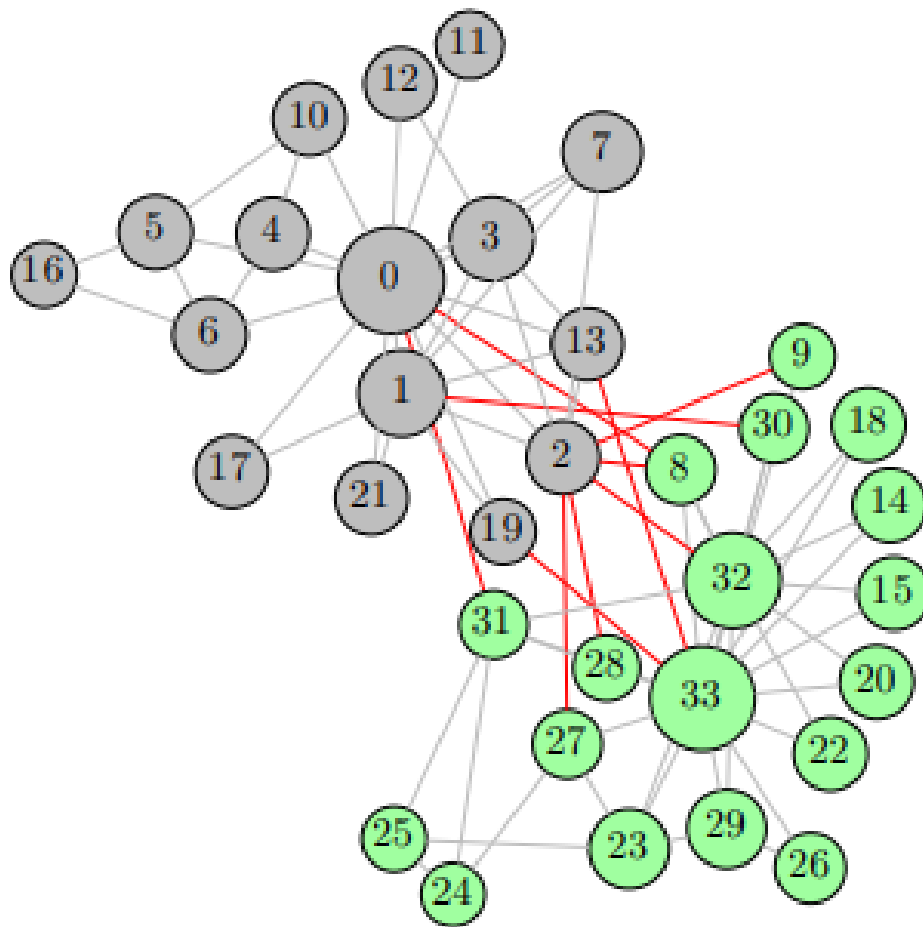


Figura 3.21: Figura retirada do artigo da ferramenta Igraph [9]. Refere-se a correta identificação de duas comunidades na rede *Zachary karate-club*. As linhas em vermelho são as arestas que se ligam entre as duas comunidades. Esse foi o resultado da aplicação do Algoritmo de localização de comunidade de Newman, através do Igraph.

Desenvolvimento de plugins e scripts

Existem duas linhas de raciocínio para adicionar funcionalidades ao Igraph. Como a ferramenta é *open source*, existe a possibilidade de incluir novas funcionalidades diretamente ao Igraph, para que fique disponível para toda a comunidade. No GitHub do Igraph é possível abrir uma *issue* para tal, ou, verificar as *issues* existentes em busca de alguma que se resolvida, seria a solução do problema. A outra maneira é apenas criar um *script* na linguagem desejada e integrar com uma das versões do Igraph, como Python ou C. O desenvolvimento de novas funcionalidades depende de qual linguagem foi adotada para o Igraph. Todas as funcionalidades já existentes estão disponíveis na ferramenta, não existe uma loja de *plugins* ou algo do tipo.

Formatos aceitos para importação e exportação

O Igraph, apesar de possuir versões para diversas linguagens (R, Python, C/C++ e Mathematica), tem uma consistência quando se trata de formatos de arquivos para importação e exportação. Os formatos aceitos para importação e exportação são: GEXF, graphML, pajek e lista de arestas.

Capítulo 4

Um estudo comparativo de ferramentas à luz de um case

4.1 Case CICFriend e addon "Badges" para acreditação de mérito interacionista

O CICFriend é um projeto de rede social descentralizada desenvolvido pelo Departamento de Ciência da Computação da Universidade de Brasília (CIC/UnB). O projeto tem como objetivo criar uma plataforma para a interação e colaboração entre estudantes, professores e pesquisadores da área de Ciência da Computação. O *addon badges* para acreditação de mérito interacionista é uma extensão para o CICFriend que permite a atribuição de *badges* digitais aos usuários da rede social. Os *badges* são concedidos com base no envolvimento dos usuários em atividades de aprendizagem informal, como participar de conversas, responder perguntas, criar conteúdo e compartilhar conhecimento. Os *badges* digitais são uma forma de reconhecer e valorizar a aprendizagem informal. Eles fornecem aos usuários um *feedback* tangível sobre seu progresso, o que pode motivar os discentes a participar mais ativamente de atividades de aprendizagem informal.

No contexto do CICFriend, os *badges* digitais são concedidos de acordo com os seguintes critérios: participação em conversas, responder e interagir com perguntas, criação de conteúdo e compartilhamento de conhecimento, como visto no estudo realizado por Égler [11]. Dessa forma, o objetivo dos *badges* é incentivar a participação dos discentes nas conversas sobre disciplinas na rede social CICFriend. Sendo assim, os estudantes precisam de uma forma de visualizar suas interações na rede para que possam compreender melhor quais são as relações que estão sendo estabelecidas com a interação na rede. Com isso em mente, a implantação de uma ferramenta de SNA na rede social se faz necessária para o alcance desses objetivos.

4.2 O comparativo proposto

Como visto no capítulo anterior, as ferramentas apresentadas possuem vários diferenciais, várias funções para trabalhar com análise de redes sociais e trabalham com vários formatos de dados, mas, alguns outros pontos fundamentais precisam ser levados em consideração para que as ferramentas possam ser devidamente utilizadas no contexto do ecossistema que ela está inserida. Apesar da escolha de uma ferramenta para análise de redes sociais não ser definitiva para o projeto, é de grande importância selecionar uma que atenda às demandas específicas e atuais da rede social. A figura 4.1 a seguir mostra algumas informações sobre as ferramentas:

| Ferramenta | Visualização dos Dados | Gratuita? | Código Aberto? | Formato dos Dados | Importa e Exporta? |
|------------|------------------------|-----------|----------------|---|--------------------|
| Cytoscape | Grafo | Sim | Sim | CSV, .xlsx, GraphML, SIF, PSI-MI, XGMML, GEXF e formatos de imagem (PNG, JPEG, etc) | Sim |
| NetworkX | Grafo | Sim | Sim | JSON, Edge list, Node list, GraphML, Pajek, GEXF, NetworkX pickle format, DOT | Sim |
| SocnetV | Grafo | Sim | Sim | GraphML, GML, Pajek, DOT, matriz de Adjacência e lista de arestas | Sim |
| Gephi | Grafo | Sim | Sim | GEXF, CSV, GraphML, GML, GDF e Pajek NET | Sim |
| Graph-tool | Grafo | Sim | Sim | GEXF, GraphML, pajek, lista de arestas e LGL | Sim |
| Igraph | Grafo | Sim | Sim | GEXF, graphML, pajek e lista de arestas | Sim |

Figura 4.1: Ferramentas para análise de redes sociais e seu comparativo com relação ao necessário para atuar no projeto.

A figura 4.1 mostra algumas características essenciais das ferramentas para que pudessem ser levadas em consideração como opções dentro do projeto. Os dois fatores mais importantes são a gratuidade de uso da ferramenta, já incluindo licenças de uso que permitam o seu acoplamento dentro do projeto do ecossistema e também em qualquer outra instância do Friendica. O segundo ponto fundamental para a implantação de quaisquer ferramentas é o código aberto. Softwares de código aberto oferecem vantagens, como

transparência, flexibilidade e comunidade ativa de desenvolvedores. Dessa forma, é possível ter mais poder sobre a ferramenta, permitindo personalizações específicas para as necessidades do projeto. Além disso, a comunidade pode contribuir com aprimoramentos e correções mais rapidamente. Todas as ferramentas que foram trazidas para o trabalho contemplam esses dois pontos fundamentais. Um terceiro ponto também importante mas não decisivo para a escolha de uma ferramenta seria o formato dos dados que é possível de se trabalhar. Esse ponto é importante pois a ferramenta de SNA precisaria conseguir receber os dados do Friendica. A rede social consegue exportar dados para os formatos CSV, JSON, formatos conhecidos e com relativa facilidade de se obter compatibilidade em outras ferramentas.

Para apoiar a escolha de uma dessas ferramentas, 5 pontos foram considerados. São eles:

- **Recursos e funcionalidades:** É importante validar esse ponto pois, apesar das ferramentas serem escolhidas para trabalhar com SNA, é necessário que a ferramenta ofereça recursos e funcionalidades que atendam às necessidades do projeto. Como exemplo de recursos que seriam importantes estarem presentes nas ferramentas, podemos citar: capacidade de coletar e analisar dados de redes sociais, possibilidade de visualizar e analisar grafos, suporte a diversas linguagens de consulta e capacidade de se integrar com outras ferramentas.
- **Facilidade de uso:** Por trabalhar em um contexto complexo, com um volume de informações gigantesco e sendo necessário fazer diversas análises com essas informações, é importante que a ferramenta seja de fácil uso e ofereça uma interface amigável para os usuários. Isso pode ajudar a garantir que a ferramenta seja utilizada de forma eficiente e eficaz.
- **Documentação e suporte:** A ferramenta precisa estar documentada de forma detalhada e completa, dessa forma o seu uso se torna mais fácil e provê uma melhor compreensão das suas capacidades e funcionalidades. Ter suporte disponível para ajudar os usuários a entender como utilizar a ferramenta é um ponto importante para se verificar, sempre pensando na melhora do uso e da exploração das suas capacidades.
- **Atividade da comunidade de usuários:** Verificar se a ferramenta possui uma comunidade ativa de usuários e se existem exemplos e recursos disponíveis online ou em livros para dar suporte aos usuários.

- **Desempenho e escalabilidade:** é importante verificar se a ferramenta é capaz de lidar com grandes quantidades de dados e se é escalável para atender às necessidades crescentes do projeto.

Esses pontos são importantes de serem levados em consideração para escolher uma ferramenta que ofereça os recursos e funcionalidades necessários, seja fácil de usar e tenha uma comunidade ativa de usuários, além de ser capaz de lidar com grandes quantidades de dados e ser escalável. Assim, futuros projetistas do ecossistema podem ter um apanhado de sólidas ferramentas para SNA, ao passo que conseguem diferenciar a escolha da ferramenta mais adequada não só através dos pontos indicados acima, mas também, a partir das características que tornam cada uma das ferramentas única para um contexto mais específico.

4.3 Redesenho do addon Badges incluindo visualização das interações

Para se ter uma base de como seria a visualização de interações através da ferramenta selecionada pelo projetista e inserida dentro do Friendica, algumas imagens foram montadas usando a ferramenta Cytoscape como exemplo. Como a ideia inicial é que os usuários possam ter uma maneira de verificar suas interações através de um grafo e, a partir dele, extrair informações sobre a obtenção de *badges* digitais por exemplo, essa seção pode estar no menu superior ou pode ser criado um menu específico com um conjunto de ações voltados para a visualização das interações.

Dentro do menu de interações, o usuário pode interagir com o grafo, configurando-o, aplicando alguns filtros e pode verificar suas interações dentro da rede social. A exemplo, temos um usuário que está prestes a receber 2 *badges*, que estão relacionadas ao comportamento do usuário. A partir disso, muitos aspectos serão analisados, sendo mostrados apenas aqueles que mostram ao usuário o que ele está prestes a obter. Outra sugestão é que tenha um menu lateral, onde o usuário poderá escolher as ações dentro do menu de interações. A exemplo, o menu de ações contempla as opções:

1. **Grafo de interações:** mostra o grafo gerado pela ferramenta, além de mostrar, na sua lateral, as *badges* perto de ser obtidas.
2. **Minhas badges:** mostra as *badges* já obtidas pelo usuário em uma seção. Em outra seção, chamada "a obter", estarão listadas todas as *badges* possíveis do usuário conseguir através das interações.



Figura 4.2: Protótipo do Friendica com a análise de interações no menu do usuário.

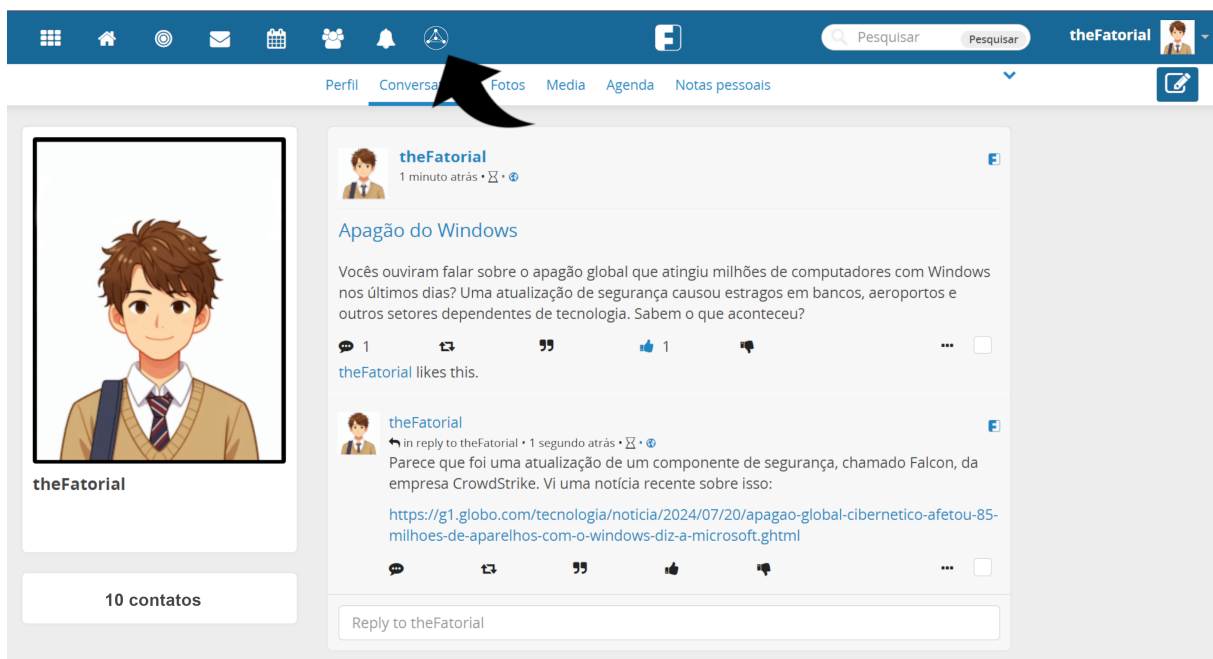


Figura 4.3: Protótipo do Friendica com a análise de interações no menu superior.

3. **Minhas interações:** mostra mais detalhes sobre as interações feitas 1 a 1, com dados obtidos através da análise da ferramenta de acordo com as interações do usuário atual.

4. **Configurar grafo:** mostra todas as configurações possíveis de se fazer no grafo, que serão extraídas da ferramenta a princípio.

O esquema pode ser verificado na imagem a seguir.

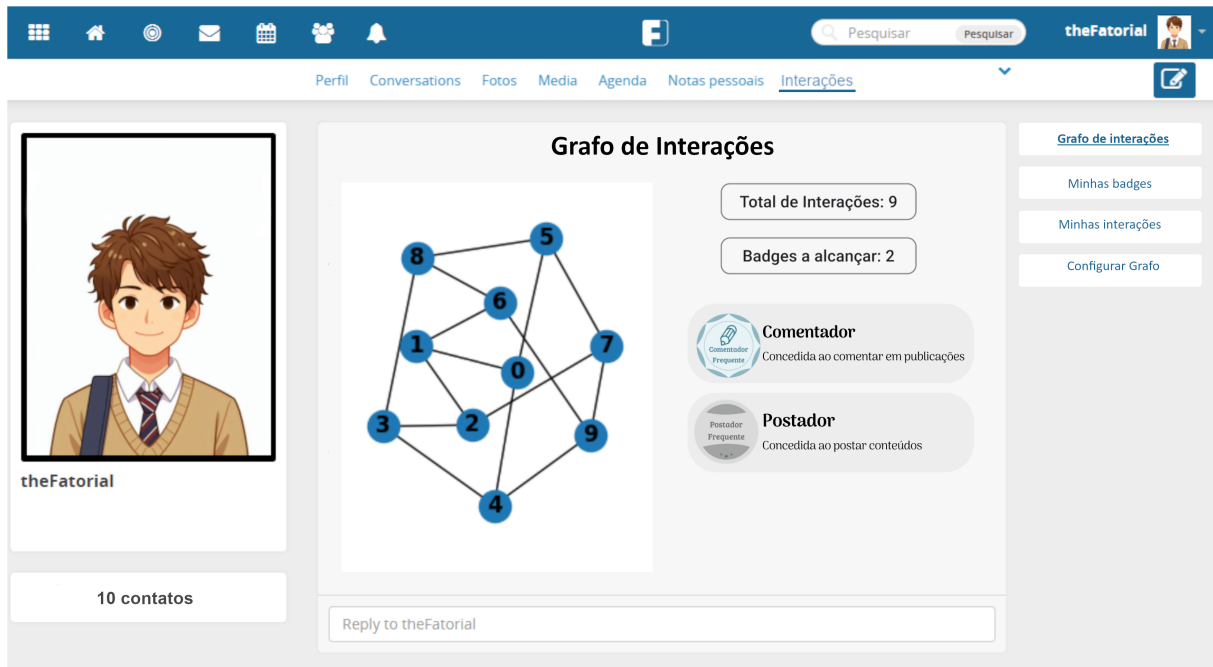


Figura 4.4: Protótipo do Friendica com a análise de interações no menu superior.

Outra proposta é a de em vez de mostrar apenas o grafo vindo da ferramenta de visualização, mostrar mais seções da ferramenta, como, por exemplo, a área de configurações. Essa área de configurações poderia ser acionada por um menu lateral, posicionado do lado direito e em tamanho reduzido, para que a atenção principal do usuário esteja na ferramenta. Um ponto positivo nesse tipo de abordagem é a maior independência do usuário para configurar seu grafo da maneira que preferir. Porém, seria inserida uma complexidade um pouco maior para o público geral que não está acostumado a parâmetros e termos mais técnicos, então essa opção expansiva da ferramenta deve ser tratada com cuidado. Uma sugestão seria colocar dois modos de configurações: avançado e simples. O modo simples abriria apenas opções de estilização de nós e arestas, tamanhos, formas, grau de proximidade, zoom e formato do grafo. Já no modo avançado, todas as configurações do grafo seriam mostradas, dando a possibilidade do usuário realizar ajustes e estilizações mais avançadas, como por exemplo com a opção de aplicar algoritmos, realizar análises específicas, entre outras várias possibilidades. Essa proposta ainda prevê um botão específico para acesso de extensões. Aqui, existem as opções de colocar a disposição do usuário apenas as extensões pré-aprovadas e desenvolvidas especificamente para serem usadas no

ecossistema, ou, abrir para que o usuário acesse todas as extensões que a ferramenta pode oferecer, que, a depender da ferramenta, pode haver diversas opções que foram produzidas pela comunidade. Nesse último caso, cabe atenção especial pois podem existir extensões com códigos mal intencionados, que podem comprometer aspectos de segurança da rede.

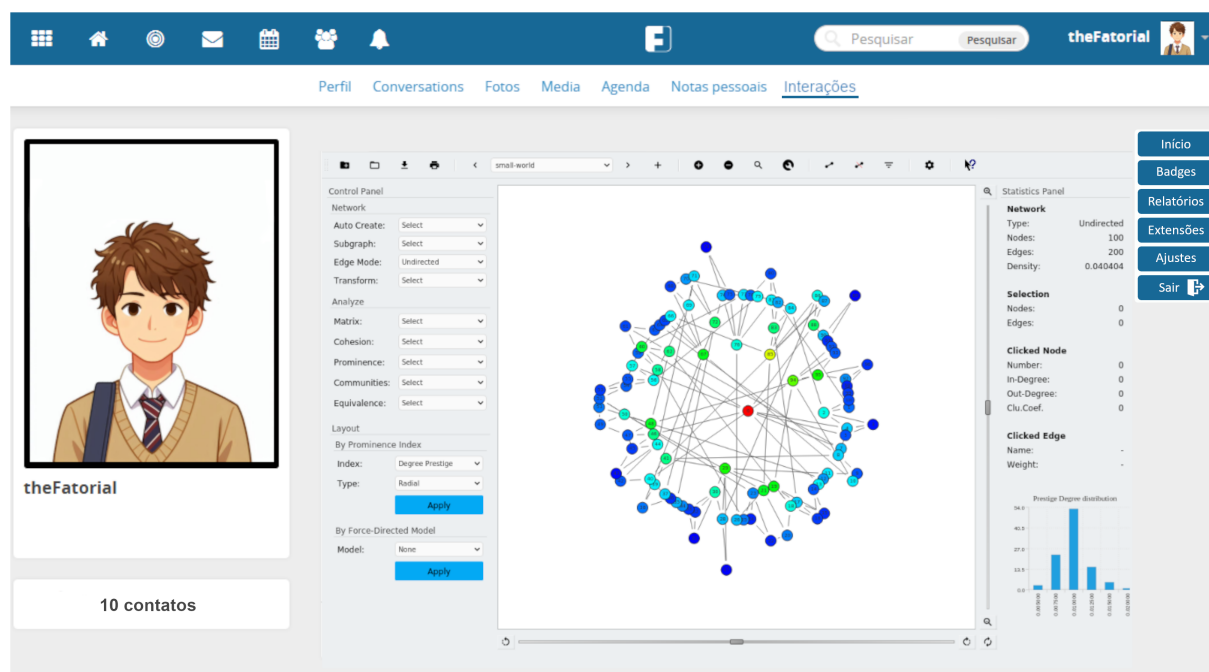


Figura 4.5: Protótipo da área de interação em modo *fullscreen* na seção de configurações da ferramenta.

O menu da ferramenta pode contar com uma área para *badges*, com o intuito de reunir em um lugar só as informações que se relacionam com SNA e visualização de interações. Então, o usuário tem a possibilidade de estar a par tanto do grafo de interações, quanto das *badges* obtidas e o quão perto está de obter cada uma delas. As *badges* podem ter um papel importante na vida acadêmica, pois, validadas e consolidadas no meio acadêmico, podem agregar valor para o aluno e podem servir como um certificado de que o usuário executou determinada atividade dentro do contexto acadêmico que estiver inserido. As *badges* podem ser fornecidas por toda entidade na instituição, como uma forma de fomentar certas atividades e agregar interesse coletivo nessas atividades. Assim como Égler [11] menciona em seu trabalho, vários aspectos podem ser analisados com as *badges* como sinais de aprendizado formal e informal. Existe um leque grande de possibilidades que podem ser abordadas unindo a obtenção de *badges* com a análise de interações, aspectos que serão abordados no capítulo posterior.

Por fim, temos a área de relatórios. Ao clicar no botão de relatórios, os usuários terão acesso a uma visão aprofundada de suas atividades na plataforma. Esse botão além de

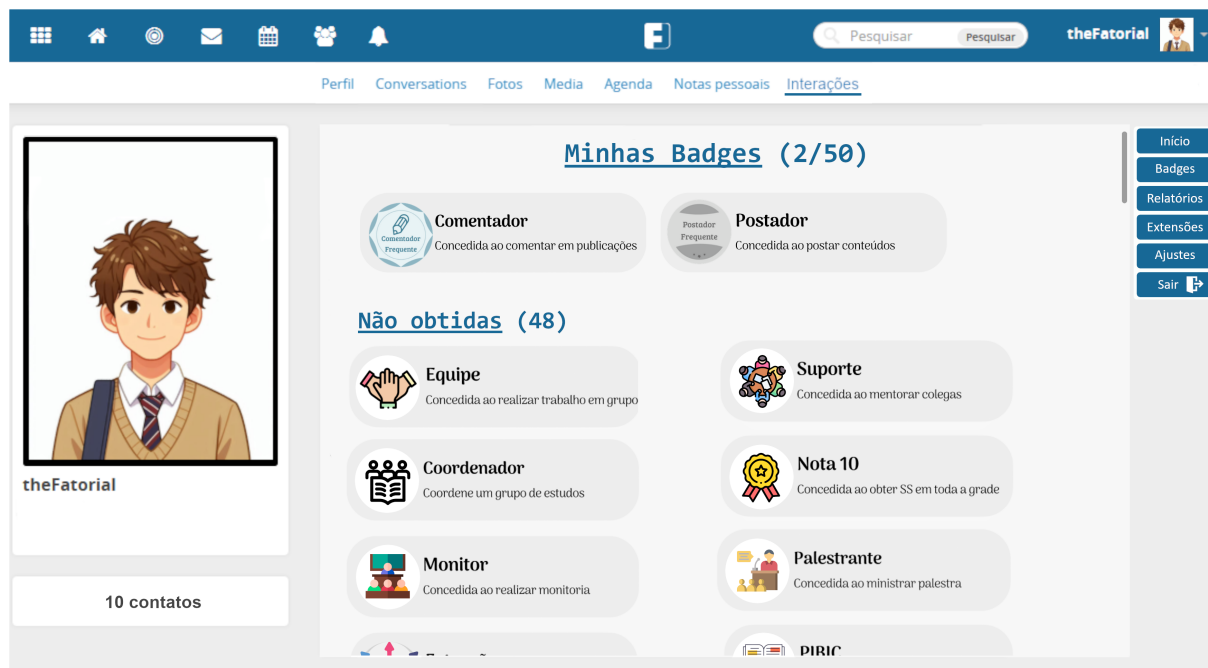


Figura 4.6: Protótipo da área de *badges* no menu superior de interações.

gerar relatórios personalizados, abrangendo períodos diários, mensais, semestrais e anuais, também armazena os relatórios feitos anteriormente, para que seja possível o usuário ter um histórico de suas atividades na rede e o seu progresso de acordo com o tempo. Os dados apresentados incluem a quantidade de interações realizadas, *badges* conquistados e aquelas próximas de serem obtidas, além de *insights* valiosos a partir das interações desse usuário dentro da rede social. Essa funcionalidade permite que os usuários acompanhem seu crescimento na plataforma, identifiquem oportunidades de interação e personalizem sua experiência no Friendica.

Capítulo 5

Conclusão

5.1 Objetivos alcançados

O panorama de ferramenta fornece um comparativo profundo a respeito das ferramentas de SNA que seguem os critérios estabelecidos anteriormente no capítulo 1. Tais características fazem respeito ao padrão de implantação do ecossistema SmartUnB.ECOS, que prevê uma rede social descentralizada, com código aberto e gratuita. As mesmas características devem ser seguidas pela ferramenta de SNA que será implantada no ecossistema.

O panorama de ferramentas mostra, de maneira indireta, dois tipos de ferramentas de SNA: com *layout* pronto e sem *layout* pronto. As ferramentas com *layout* pronto, são aquelas que, além das ferramentas internas e funcionalidades, contam com uma interface para que o usuário possa interagir diretamente com elas. Esse é o caso da Cytoscape, SocnetV e Gephi. As outras ferramentas são aquelas que não contam com uma interface de usuário. Essas ferramentas tem todas as suas funções e funcionalidades acionadas via linha de código exclusivamente. Esse é o caso da NetworkX, Graph-tool e Igraph. A interface é um detalhe que pode passar despercebido, mas faz muita diferença na hora da implantação, pois aquelas ferramentas que contam com a interface tem uma aplicação mais direta, com poucos ajustes na hora de adequar ao ecossistema, pois as suas funções já estão agrupadas e acessíveis através da interface do usuário. Esse não é o caso das ferramenta que tem seu acionamento exclusivamente através da linha de código, pois para que as interações possam acontecer, é necessário programar lógicas e atribuí-las a botões em uma interface que deve ser previamente pensada para que seja intuitiva para o usuário final.

Apesar desses pontos, não necessariamente ferramentas sem interface gráfica são menos vantajosas para implantação em um projeto com o ecossistema. Deve-se olhar para os objetivos que se querem com a ferramenta e o prazo para a implantação. Ferramentas sem interface devem considerar planejar e executar a produção do *layout* do usuário. A

vantagem que se tem na produção de uma interface própria é a capacidade de estilizar e configurar todos os aspectos da interface, posicionamento e acionamento de funcionalidade para a ferramenta em questão. É possível prever integrações futuras de outras ferramentas e, assim, preparar o *layout* ou até mesmo a ferramenta para uma integração. Por conta das características de código aberto e gratuidade de uso, é possível realizar ações como essa.

Outro ponto importante de salientar é que duas ferramentas possuem uma versão *web* (Cytoscape e Gephi). Isso possibilita uma rápida injeção de código *embed* na instância do Friendica e em outros projetos que necessitem de visualização. Com passagem dos dados a visualização é garantida pela versão *web* das ferramentas.

É importante considerar também o desempenho das ferramentas frente a grandes conjuntos de dados. É fácil imaginar que a implantação da rede social Friendica gerará muitas interações devido a quantidade de usuários e a quantidade de afazeres de cada um deles e que os levam diretamente para a rede. A quantidade de nós pode variar de milhares a milhões de nós. Sendo assim, ferramentas com foco em desempenho e escalabilidade seriam adequadas para implantação no ecossistema, como é o caso da Graph-tool e Igraph.

O objetivo geral deste trabalho era avaliar e comparar ferramentas de visualização de redes sociais que tenham como características o código aberto, gratuidade para uso e que realizem a importação e exportação de dados, com o objetivo de auxiliar a escolha de uma ferramenta para a visualização de redes sociais no ecossistema educacional SmartUnB.ECOS, especificamente na instância Friendica (CICFriend). Para tanto, foi fornecida uma avaliação de um panorama de 6 ferramentas de SNA, mostrando suas principais capacidades, para que, após avaliação dos requisitos que devem ser cumpridos, uma das ferramentas possa ser selecionada para compor a arquitetura do ecossistema.

5.2 Considerações finais

O trabalho de Égler [11] abre um leque de possibilidades para o uso de ferramentas de visualização de interações dos estudantes em redes sociais descentralizadas. No contexto da CICFriend, os cenários potenciais de visualização podem ser explorados em três áreas principais: disciplinar, colaborativa e de impacto.

Na área disciplinar, as visualizações podem ser usadas para identificar estudantes que estão se saindo bem ou mal em uma determinada disciplina, ou que estão se conectando com outros estudantes que compartilham seus interesses. Por exemplo, uma visualização de participação em conversas por disciplina poderia mostrar que um determinado grupo de estudantes está participando mais das conversas sobre uma determinada disciplina do

que outros grupos. Isso poderia indicar que esses estudantes estão se saindo bem na disciplina ou que estão interessados em aprender mais sobre ela.

Na área colaborativa, as visualizações podem ser usadas para identificar redes de colaboração entre estudantes. Por exemplo, uma visualização de redes de colaboração entre estudantes poderia mostrar que um determinado estudante está colaborando com um grupo de estudantes de diferentes departamentos ou de diferentes universidades. Isso poderia indicar que esse estudante está aberto a ensinar e aprender com pessoas de diferentes *backgrounds*. Além disso, pode-se identificar se padrões como esse se repetem em outros contextos da universidade, dentro da rede social. A visualização de interações também podem ser usadas para identificar comunidades de prática ou grupos de interesse entre estudantes. Essas informações podem ser usadas para apoiar o desenvolvimento de programas e atividades que atendam às necessidades desses grupos. Por exemplo, uma visualização de interações entre estudantes poderia mostrar que esse grupo está interessado em desenvolvimento de software, o que forneceria respaldo para justificar que há uma demanda por cursos ou atividades sobre esse tema.

Na área de impacto, as visualizações podem ser usadas para entender como as interações dos estudantes estão contribuindo para a sua aprendizagem e desenvolvimento. No contexto da CICFriend, as visualizações podem ser usadas para avaliar o impacto das interações dos estudantes em sua aprendizagem formal e informal, identificar padrões de interação que estão associados a resultados positivos de aprendizagem e desenvolver estratégias para melhorar o impacto das interações dos estudantes na aprendizagem. Por exemplo, com a visualização das interações dos estudantes na aprendizagem formal, poderia ser mostrado que os estudantes que interagem mais com os professores têm um desempenho melhor nas provas, o que poderia indicar que as interações com os professores são importantes para o aprendizado dos estudantes. Outra possibilidade seria identificar padrões de interação que estão associados a resultados positivos de aprendizagem informal. Por exemplo, uma visualização das interações dos estudantes na aprendizagem informal poderia mostrar que os estudantes que participam de comunidades de prática têm um maior nível de satisfação com a sua aprendizagem. Isso poderia indicar que as comunidades são uma forma eficaz de promover a aprendizagem informal.

Além dos cenários potenciais mencionados, existem outras possibilidades de visualização que poderiam ser exploradas no contexto da CICFriend ou da UnBFriend. Por exemplo, as visualizações poderiam ser usadas para:

- Identificar líderes e influenciadores
- Visualizar o fluxo de informações
- Identificar padrões de comportamento

Como mencionado no capítulo 3, existem diversos algoritmos e funções comuns a todas as ferramentas, que podem ser usadas para identificar os itens listados acima. A título de exemplo, pode-se citar a centralidade de grau, que indica a quantidade de conexões diretas que um nó possui. Usuários com alto grau tendem a ser mais influentes, pois possuem um grande número de contatos. A centralidade de *betweenness* mede a importância de um nó como intermediário em caminhos mais curtos entre outros nós. Aqueles com alta *betweenness* ocupam posições estratégicas na rede e podem controlar o fluxo de informações. A análise de comunidades permite identificar grupos de nós densamente conectados entre si e fracamente conectados a outros grupos. Essas comunidades podem representar diferentes grupos sociais, profissionais ou temáticos. A análise de caminhos identifica os caminhos mais curtos entre dois nós, revelando as rotas mais eficientes para a disseminação de informações. Essas funções podem ser usadas para identificar as características listadas anteriormente para exploração no contexto da CICFriend.

O uso de ferramentas de visualização de SNA pode proporcionar *insights* valiosos sobre a participação dos estudantes em redes sociais descentralizadas. Esses *insights* podem ser usados para melhorar a experiência de aprendizagem dos estudantes e para promover a aprendizagem informal. Critérios como esse podem ser usados para gerar uma variedade de cenários potenciais de visualização no CICFriend.

Tendo em vista os detalhes mencionados em relação ao CICFriend, pensando em um contexto maior, as possibilidades seriam bem mais abrangentes, como por exemplo, levar em consideração a UnBFriend. O CICFriend abrange apenas o departamento de ciência da computação da UnB, no entanto, a UnBFriend seria uma rede social maior e mais diversificada, o que criaria novas possibilidades de visualização. Por exemplo, seria possível visualizar a participação dos estudantes em conversas por campus, por curso ou por área de estudo. Também seria possível visualizar as redes de colaboração entre estudantes de diferentes departamentos ou de diferentes universidades. Além disso, seria possível visualizar o impacto dos *badges* digitais na aprendizagem dos estudantes em diferentes níveis de ensino, como graduação, pós-graduação ou extensão. A escolha de uma ferramenta de visualização de interações dos estudantes (SNA) deve levar em consideração os cenários potenciais de visualização que serão usados. A ferramenta deve ser capaz de gerar visualizações que sejam informativas, relevantes e fáceis de entender. Mesmo que o presente trabalho tenha selecionado uma ferramenta para atender ao CICFriend, nada impede que as demais ferramentas possam ser acopladas ao ecossistema, adicionando ou substituindo a escolha feita.

5.3 Trabalhos Futuros

Diante da análise das ferramentas frente ao projeto e suas necessidades, é evidente que há diversas oportunidades para aprofundar e expandir esta pesquisa. Para a implantação de uma das ferramentas de SNA dentro do projeto do ecossistema, aspectos éticos e de privacidade devem ser levados em consideração. Sendo assim, um trabalho futuro seria uma pesquisa de aspectos éticos relativos à implantação e a análise de dados dos usuários por ferramentas de SNA. O tema de ética é fundamental para uma implantação que siga todas as diretrizes e normas relacionados à proteção de dados. O estudo presente no trabalho [26] relata a importância do apoio a desenvolvedores e pesquisadores com relação aos desafios éticos presentes na construção de softwares que interagem com pessoas, coletam dados e os processam para fazer análises. Outra possibilidade seria a implementação e avaliação de uma das ferramentas de SNA na rede social Friendica. Além disso, seria interessante avaliar o impacto da ferramenta escolhida na dinâmica das interações sociais e sua efetividade ao analisar as interações dos usuários para acreditação das *badges* digitais.

Em consonância com elementos da arquitetura do ecossistema [1] e com os trabalhos desenvolvidos anteriormente, um estudo muito promissor seria a integração da ferramenta de SNA com sistemas de gestão de aprendizagem (LMS), com o intuito de melhorar a análise de dados educacionais. Além disso, avaliar como uma inteligência artificial (IA) pode ser usada em conjunto com a SNA para recomendar conteúdos e recursos educacionais personalizados para cada usuário, com base em suas interações na rede social. Outra recomendação é o estudo de uma tutoria inteligente, que utilize a análise de redes sociais para adaptar o processo de ensino e aprendizagem às necessidades individuais de cada aluno. Além disso a tutoria poderia identificar alunos com dificuldades de aprendizado ou em risco de evasão, com base em suas interações na rede social e em dados acadêmicos. Esse estudo é importante pois existem cursos com uma taxa de evasão muito alta.

A crescente complexidade das redes sociais e a necessidade de extrair *insights* valiosos a partir de seus dados impulsionam a demanda por ferramentas de análise visual robustas e flexíveis, principalmente pensando na inserção dentro do ecossistema. Neste contexto, a proposição de uma plataforma unificada que seja compatível com os princípios LTI (Learning Tools Interoperability) [27] e SOLID, e que possa ser consumida por diferentes plataformas de redes sociais (como Friendica e Kirilândia), representaria um avanço significativo no projeto com relação à análise de dados. Uma plataforma de análise visual de redes sociais, construída com base nos princípios LTI e SOLID, oferece vantagens como a integração *seamless* da plataforma com diversas plataformas de aprendizagem, facilitando a análise de dados sociais no contexto educacional.

Referências

- [1] Nóbrega, Germana M da, Gabriel T da Silva e Thiago VR Silva: *Um projeto estruturante para orientações de tcc em cursos de computação: que oportunidades para ihc?* Em *Anais do XIII Workshop sobre Educação em IHC*, páginas 19–24. SBC, 2022. ix, 3, 57
- [2] Nascimento, C., Carla Freitas, Renata Galante, Luciana Nedel e Jesús Mena-Chalco: *Visualização interativa de redes sociais: um estudo de caso em redes de colaboração científica*. setembro 2011. ix, 4, 10, 12
- [3] *Cytoscape: An open source platform for complex network analysis and visualization*. <https://cytoscape.org/>. Acessado: 30-11-2021. ix, 18, 19, 22
- [4] *Socnetv - social network analysis and visualization software*. <https://socnetv.org/>. Acessado: 13-11-2023. ix, 26, 27, 28, 29
- [5] *Gephi - makes graphs handy*. <https://gephi.org/>. Acessado: 14-12-2023. ix, x, 31, 32, 33, 34, 35, 38
- [6] *Gephi lite*. <https://gephi.org/gephi-lite/>. Acessado: 14-12-2023. x, 35, 36
- [7] *Graph-tool - efficient network analysis*. <https://graph-tool.skewed.de/>. Acessado: 14-01-2024. x, 38, 39
- [8] *Igraph - network analysis software package*. <https://igraph.org/>. Acessado: 14-02-2024. x, 40, 41
- [9] Csardi, Gabor e Tamas Nepusz: *The igraph software package for complex network research*. InterJournal, Complex Systems:1695, novembro 2005. x, 42, 43
- [10] Marteleto, Regina Maria: *Análise de redes sociais - aplicação nos estudos de transferência da informação*. Programa de Pós-Graduação em Ciência da Informação MCT/IBICT - UFRJ/ECO, 2001. 1
- [11] Egler, Pedro H. P.: *Agregando a aprendizagem informal à formal: Badges digitais para participação discente nas conversas em rede social descentralizada sobre disciplinas*. Monografia (Graduação em Engenharia de Computação). Universidade de Brasília (UnB), 2021. 1, 5, 45, 51, 54
- [12] Nóbrega, Germana e Fernando Cruz: *Rumo a um ecossistema educacional apoiado por computador e socialização em rede descentralizada*. Em *Anais Estendidos do XVI Simpósio Brasileiro de Sistemas Colaborativos*, páginas 36–41, Porto Alegre,

- RS, Brasil, 2021. SBC. https://sol.sbc.org.br/index.php/sbsc_estendido/article/view/16033. 2, 3
- [13] Kawagoe, Akemi Leandra: *O que aprendemos em silêncio : aprendizagem informal e ecossistemas de aprendizagem*. Dissertação de Mestrado, Universidade de Brasília (UnB), 2019. 2
 - [14] Oliveira, Luciana e Álvaro Figueira: *Analysing relevant interactions by bridging facebook and moodle*. março 2016. 2, 13
 - [15] Oliveira, Jéssica S., Germana M. Da Nóbrega, Fernando W. Cruz e Roberta B. Oliveira: *Decentralized social network for the campus: historical claims meet contemporary needs*. Em *2021 XVI Latin American Conference on Learning Technologies (LACLO)*, páginas 408–415. IEEE, 2021. 3
 - [16] Moissa, Barbara, Eduardo José de Borba, Avanilde Kemczinski e Isabela Gasparini: *Uma ferramenta de visualização da informação para analisar o comportamento do aluno em um ambiente e-learning e sua trajetória de aprendizagem*. InfoDesign - Revista Brasileira de Design da Informação, 11(3):337–351, 2014. <https://www.infodesign.org.br/infodesign/article/view/318>. 8
 - [17] Stasko, John: *Visualization for information exploration and analysis*. Em *2008 IEEE Symposium on Visual Languages and Human-Centric Computing*, páginas 7–8, 2008. 8
 - [18] Nagulendra, Sayooran e Julita Vassileva: *Understanding and controlling the filter bubble through interactive visualization: A user study*. HT 2014 - Proceedings of the 25th ACM Conference on Hypertext and Social Media, setembro 2014. 12
 - [19] Wang, Yinying e Min Kim: *Identify learning leaders in collaborative learning: Using social network analysis to develop and compare two methodological approaches*. julho 2019. 13
 - [20] *Stanford Large Network Dataset Collection*. <http://snap.stanford.edu/data/index.html>. Acessado: 03-01-2022. 14
 - [21] *Stanford Large Network Dataset Collection social circles: Twitter*. <http://snap.stanford.edu/data/ego-Twitter.html>. Acessado: 03-01-2022. 14
 - [22] *Cytoscape - source code*. <https://github.com/cytoscape/cytoscape>. Acessado: 30-11-2021. 18
 - [23] Otasek, D. et al.: *Cytoscape automation: empowering workflow-based network analysis*. Genome Biol, 2019. 22
 - [24] *Networkx — networkx documentation*. <https://networkx.org/>. Acessado: 08-12-2023. 23
 - [25] Bastian, Mathieu, Sebastien Heymann e Mathieu Jacomy: *Gephi: An open source software for exploring and manipulating networks*, 2009. <http://www.aaai.org/ocs/index.php/ICWSM/09/paper/view/154>. 30

- [26] Rangel, Bárbara Varanda: *Contribuições para conduta ética em três momentos na pesquisa em tecnologia educacional: implantação, projeto e avaliação*. 80 f., il., 2021. Trabalho de Conclusão de Curso (Bacharelado em Ciência da Computação)—Universidade de Brasília, Brasília, 2021. 57
- [27] Costa, Vitor Pereira Tavares da: *Implantando um provedor de serviços educacionais interoperáveis segundo o padrão IMS-LTI*. 70 f., il., 2021. Trabalho de Conclusão de Curso (Licenciatura em Computação) — Universidade de Brasília, Brasília, 2021. 57