UNIVERSIDADE SALVADOR - UNIFACS
PROGRAMA DE PÓS-GRADUAÇÃO EM SISTEMAS E COMPUTAÇÃO
MESTRADO EM SISTEMAS E COMPUTAÇÃO

ELMO LIGUORI CRUZ

SNSANALYSER:
UMA FERRAMENTA PARA EXTRAÇÃO E ANÁLISE DE REDES SOCIAIS A PARTIR DE COMUNIDADES EXISTENTES EM SITES DE RELACIONAMENTO

Salvador
2008

ELMO LIGUORI CRUZ

SNSANALYSER:
UMA FERRAMENTA PARA EXTRAÇÃO E ANÁLISE DE REDES SOCIAIS A PARTIR DE COMUNIDADES EXISTENTES EM SITES DE RELACIONAMENTO

Dissertação apresentada ao Curso de Mestrado Profissional em Sistemas e Computação, Universidade Salvador - UNIFACS como requisito parcial para obtenção do título de “Mestre”.

Orientador
Prof. Dr. Manoel Gomes de Mendonça Neto

Salvador
2008

FICHA CATALOGRÁFICA
(Elaborada pelo Sistema de Bibliotecas da Universidade Salvador - UNIFACS)

Cruz, Elmo Liguori
SNSAnalyser: uma ferramenta para extração e análise de redes sociais a partir de comunidades existentes em sites de relacionamento/Elmo Liguori Cruz. - 2007.
136 f.

Dissertação (Mestrado) - Universidade Salvador – UNIFACS. Mestrado em Sistemas de Computação, 2008.

Orientador: Prof. Dr. Manoel Gomes de Mendonça Neto

1. Informação - recuperação. 2. Mineração na Web 3. Redes Sociais - análise. I. Mendonça Neto, Manoel Gomes de, orient. II. Título.

CDD: 005.74068

TERMO DE APROVAÇÃO

ELMO LIGUORI CRUZ

SNSANALYSER:
UMA FERRAMENTA PARA EXTRAÇÃO E ANÁLISE DE REDES SOCIAIS A PARTIR DE COMUNIDADES EXISTENTES EM SITES DE RELACIONAMENTO

Dissertação aprovada como requisito parcial para obtenção do grau de Mestre em Sistemas e Computação da Universidade Salvador – UNIFACS, pela seguinte banca examinadora:

Manoel Gomes de Mendonça Neto – Orientador _____
Ph.D. in Computer Science, University of Maryland at College Park, 1997, EUA
Universidade Salvador - UNIFACS

Cesar Augusto Camillo Teixeira _____
Pós Doutor em Teleinformática, University of Kent at Canterbury, CANTERBURY, Inglaterra - 1994
Universidade federal de São Carlos - UFSCAR

Celso Alberto Saibel Santos _____
Docteur en Informatique, Université Paul Sabatier de Toulouse, 1999, França Universidade
Universidade Salvador - UNIFACS

Salvador, 03 de outubro de 2008

AGRADECIMENTOS

Em primeiro lugar, agradeço a Deus, como não podia ser diferente. Sou católico e acredito nas boas ações. Aprendi isso com minha família, que é o espírito Dele presente em minha vida. Agradeço pela minha vida, pela minha saúde e por ser feliz.

Aos meus pais, ELMO e ANGÉLICA, agradeço por me formarem no sentido mais amplo da palavra. O que sou hoje devo aos genes que vocês me deram e à educação que, com tanta dedicação, não poderia ter sido melhor. Aproveito este momento para dizer que vocês foram e sempre serão os grandes MESTRES da minha vida.

À minha esposa, DANIELA, agradeço por me amar, me aceitar como sou e a tornar minha vida mais feliz. Sem você, estudando ao meu lado, seria muito difícil abrir mão de meus fins de semana no cumprimento dessa etapa de minha vida.

A minhas irmãs, ÂNGELA E LUCIANA, amigas e verdadeiramente irmãs, agradeço por serem irmãs, por serem mães, por se preocuparem comigo e, sobretudo, por sempre terem acreditado no meu potencial. Vocês me fizeram acreditar que eu era capaz.

Aos meus avós que adorariam estar presentes nessa conquista, agradeço por terem sido avós, por terem me estragado e dado trabalho para meus pais me educarem, com vocês aprendi o conceito de avó e de como é divertida a companhia das crianças.

Agradeço também aos meus maravilhosos sobrinhos, JOÃO, PEDRO e NATÁLIA, por revitalizarem minhas energias com seus sorrisos sinceros e suas deliciosas brincadeiras de criança. Amo vocês.

Aos meus amigos, cunhados, padrinhos, sogro, sogra, madrasta, tios e primos, agradeço por ampliarem minha família, compreenderem minha ausência e por me proporcionarem momentos de alegria e descontração nas horas em que eu precisava rir para não chorar.

Agradeço aos professores que tive e às universidades por onde passei - UFBA e UNIFACS - que me ensinaram a aprender. Em especial agradeço ao meu orientador MANOEL GOMES DE MENDONÇA NETO pelo apoio e incentivo ao longo do Mestrado. Com você, aprendi lições que vou levar por toda a vida.

À CPMBRAXIS, agradeço pelo apoio no início dessa caminhada e pelo crescimento profissional que obtive durante o tempo que fiquei na empresa. Os conhecimentos obtidos nesse período foram muito importantes para essa empreitada.

À POWER, em especial a ERIC, por ter me dado todo o suporte necessário ao desenvolvimento desse trabalho. A POWER sem dúvida foi quem mais contribuiu com essa pesquisa e espero que seja quem mais vai usufruir dos resultados dela.

A todos que contribuíram direta ou indiretamente com este trabalho, muito obrigado.

RESUMO

Com o crescimento do número de sites de relacionamento, uma importante fonte de dados surgiu e tem se expandido continuamente. Encontrar uma forma de obter informações dos usuários de sites de relacionamento, com interesses ou características específicas, e como eles estão relacionados é um desafio, uma vez que a maioria deles possui uma base de milhões de usuários cadastrados. Nesse contexto, evidencia-se a necessidade de criação de soluções aptas a extrair e analisar redes sociais de sites de relacionamento. O presente trabalho descreve uma ferramenta implementada para tal fim. Além das suas características técnicas e o processo de desenvolvimento, também é apresentado um exemplo de uso da ferramenta aplicando-a na análise de redes sociais extraídas a partir de comunidades existentes em um site de relacionamento. O trabalho se fundamenta nos conceitos relacionados aos temas recuperação de informação, mineração na *web* e análise de redes sociais. A principal contribuição deste trabalho é a proposta de uma solução que possibilita explorar e analisar novas redes compostas por internautas de todo o mundo, selecionados por um perfil ou característica específica, diretamente de um site de relacionamento, sem a necessidade do uso de APIs específicas da empresa responsável pela administração do site.

Palavras-chave: Recuperação de Informação; Mineração na *Web;* Análise de Redes Sociais.

ABSTRACT

With the growth in the number of social networking sites, an important data source emerged and has been expanding continuously. Finding a way to extract relationship information from social networking sites using only the user pages and their interests or characteristics as searching criteria is a challenge, since most of those sites have databases of millions of users. In this context, it's clear the usefulness of a solution to extract and analyze social networks from relationship sites. This work describes a tool implemented for this purpose. In addition to its technical characteristics and development process, this thesis also presents a case study in which the tool is applied to the analysis of social networks extracted from existing communities of a well know social networking site. The work is based on concepts related to the themes of information retrieval, web mining and social networks analysis. The main contribution of this work is to propose a solution that allows exploring and analyzing networks composed of Internet users around the world, selected by profile or specific characteristics, directly from their social networking web pages, without the usage of specific APIs provided by the site administration companies.

Keywords: Information Retrieval; Web Mining; Social Network Analysis.

LISTA DE FIGURAS

LISTA DE QUADROS

SUMÁRIO

1 INTRODUÇÃO

Análise de redes sociais é uma área que tem despertado crescente interesse na comunidade científica. Isso pode ser verificado na comprovação feita por Otte e Rousseau (2002, p. 441-453) onde se constata que, para o período de 1974 até 2000, houve um crescimento linear do número de artigos publicados anualmente cujo assunto era análise de redes sociais.

A rede social de cada pessoa desempenha um papel muito importante. Hanneman afirma que a abordagem de rede enfatiza que o poder é inerentemente relacional. Para Hanneman (2001, p. 60), um indivíduo não tem poder no abstrato, ele tem poder, porque ele pode dominar outros. Assim, na avaliação do poder de cada indivíduo, a análise de redes sociais tem um lugar de destaque, uma vez que traz uma visão mais sistêmica quando comparada com estudos científicos tradicionais. Enquanto na abordagem tradicional assume-se que os atributos dos indivíduos é que são importantes, na análise de redes sociais esses atributos assumem um papel secundário, deixando o destaque principal para as relações entre eles dentro da rede (WASSERMAN; FAUST, 1994, p. 8).

Com o crescente número de sistemas que disponibilizam na *web* as relações sociais mantidas pelas pessoas, o interesse de pesquisadores nessa área acabou acompanhando essa tendência. A consolidação do uso de sites de relacionamento na atualidade oferece uma oportunidade ímpar de realização de estudos onde a rede social é composta por qualquer grupo de pessoas no mundo cadastrado num desses sistemas. As empresas, por exemplo, poderiam, após uma análise, escolher os usuários que apresentam um determinado perfil e possuem uma posição de destaque na rede, selecionando assim um público-alvo bastante adequado a campanhas de promoção ou direcionamento de produtos.

Neste contexto, apresenta-se o problema de como possibilitar que empresas e pesquisadores tirem proveito dessas valiosas informações sobre os usuários que existem e estão disponíveis em sites de relacionamento. Ou seja, como extrair uma rede social para interesses específicos a partir de um site de relacionamento, uma vez que os mais utilizados, de um modo geral, são compostos por milhares de usuários cadastrados?

Para extrair as informações da *web*, a solução é a utilização de programas conhecidos como *spiders* ou *web crawlers*, aplicações que funcionam como um robô acessando as páginas disponíveis na *web*, recuperando documentos e obtendo recursivamente os documentos referenciados por ele (KOSTER, 1995, p. 1). Para a construção e posterior análise de redes sociais extraídas a partir da *web*, são necessários *spiders* que permitem a extração de informações de sites de relacionamento.

Já com relação ao escopo, raras são as situações práticas em que o objeto de estudo engloba todos os usuários de um site de relacionamento. Uma característica muito comum neste tipo de sistema é a aglomeração em torno de grupos ou comunidades. Ou seja, um usuário cria uma comunidade com um tema específico e as pessoas que se identificam com aquela característica ou área de interesse se torna um membro daquele grupo. Desta forma, a utilização desses aglomerados com foco bem definido como base para o estudo da rede social se torna muito atraente, além de viabilizar estudos mais rápidos e rentáveis para as empresas. Afinal, melhor do que atingir um número grande de pessoas quaisquer é atingir as pessoas certas.

Entretanto, apenas montar a rede não é suficiente para obter informações relevantes sobre os indivíduos. Para uma melhor avaliação, a utilização de métricas aplicadas a redes sociais é fundamental. Elas ajudam a mostrar as características de cada um com relação à interação com outros e definem seu posicionamento dentro da rede. Segundo Costa e outros (2007), em redes, quanto maior o número de caminhos nos quais um nó ou aresta participa, maior a importância deste nó ou aresta para a rede. Assim, assumindo que as interações seguem os menores caminhos entre dois vértices, é possível calcular a importância de um vértice ou aresta. Para obter essas e outras informações, muitas métricas podem ser aplicadas na análise de redes sociais.

É interessante oferecer uma solução expansível com um conjunto de medidas fundamentais, mas com a possibilidade de incorporar novas medidas à análise e exportar os dados da rede para avaliação em ferramentas existentes no mercado. Desta forma, pode-se extrair e analisar de forma livre e ampla redes sociais obtidas a partir de comunidades existentes em sites de relacionamento.

1.1 OBJETIVO DA DISSERTAÇÃO

O objetivo desta dissertação é propor uma solução para extração de membros de comunidades pertencentes a sites de relacionamento, montagem da rede social composta pelos relacionamentos desses usuários entre si de acordo com os relacionamentos que eles possuem no site de relacionamento e aplicação de métricas na rede social resultante.

Para tanto, a proposta emprega mecanismos de customização e exportação para permitir que a extração seja feita em qualquer site de relacionamento. Além disso, outras métricas, que não as implementadas no sistema, podem ser aplicadas na rede resultante, através da inclusão de métricas customizadas ou exportação da rede social construída. Desta forma, possibilita-se escolher não só o site de relacionamento de onde a rede é extraída, mas também as métricas que serão aplicadas a ela.

1.2 ABORDAGEM ADOTADA

No intuito de atingir o objetivo acima descrito, foi construída a ferramenta SNSANALIZER, a qual realiza a extração de membros de comunidades pertencentes a sites de relacionamento, monta a rede social composta pelos relacionamentos desses usuários entre si e calcula métricas para permitir a análise da rede social resultante. Desta forma, deseja-se apoiar pessoas e profissionais a realizar análises de redes sociais criadas a partir de grupos específicos com interesses e características comuns existentes em sites de relacionamento.

No cumprimento deste objetivo, foi realizado um estudo sobre o processo de extração de informações da Internet e a utilização de métricas em análises de redes sociais.

É feita uma demonstração de uso da ferramenta SNSANALIZER com todas as suas funcionalidades. A fim de abordar uma aplicação prática da ferramenta, a demonstração é feita sobre comunidades do site de relacionamento Orkut ^{TM} (ORKUT, 2008).

1.3 ORGANIZAÇÃO DA DISSERTAÇÃO

Os próximos dois capítulos apresentam a fundamentação teórica e revisão da literatura relacionada ao tema. Os capítulos quatro e cinco contêm a descrição do trabalho desenvolvido nesta dissertação. O sexto capítulo conclui o trabalho e em seguida são listadas as referências

eletrônicas e bibliográficas utilizadas neste trabalho. Por fim, segue um apêndice contendo material suplementar utilizado no processo de desenvolvimento da ferramenta.

Desta forma, esta dissertação encontra-se dividida nos seguintes capítulos:

- CAPÍTULO 1 – descreve os objetivos, a abordagem e escopo da pesquisa, bem como a organização do documento do presente trabalho.
- CAPÍTULO 2 – apresenta o tema análise de redes sociais. Neste capítulo, destacam-se conceitos, métricas e algumas ferramentas de cálculo.
- CAPÍTULO 3 – aborda o tema extração automática de informações da Internet, relatando as estratégias utilizadas para obter as informações dos sites de relacionamento.
- CAPÍTULO 4 – descreve detalhadamente a ferramenta SNSAnalyser e sua implementação em Visual C# ™ (MICROSOFT, 2007).
- CAPÍTULO 5 – faz uma avaliação da aplicabilidade da ferramenta SNSAnalyser na área de extração e análise de redes sociais e apresenta um exemplo de uso da ferramenta.
- CAPÍTULO 6 – apresenta as conclusões deste trabalho relacionando suas contribuições, limitações e perspectivas futuras;
- APÊNDICE A – mostra, segundo a notação UML™, alguns diagramas elaborados ao longo do processo de desenvolvimento da ferramenta SNSAnalyser.
- APÊNDICE B – apresenta uma API detalhada do Power Script.

2 ANÁLISE DE REDES SOCIAIS

Uma rede social consiste de um ou mais conjuntos finitos de atores e todas as relações definidas entre eles. Os atores são os nós da rede e as relações entre eles são as arestas. O primeiro uso da técnica de análise de redes sociais data de 1933 quando o psiquiatra Jacob Levi Moreno apresentou o sociograma, ferramenta originária de seu trabalho em sociometria. Moreno criou o método para analisar relacionamentos emotivos interpessoais dentro de um grupo. Por meio de sua ferramenta era possível identificar líderes e indivíduos isolados (MORTON e outros, 2004).

Hanneman afirma que “a análise de redes sociais é mais um ramo da sociologia ‘matemática’ do que de ‘análise estatística ou quantitativa’, embora os estudiosos devam certamente praticar as duas abordagens” (HANNEMAN, 2001, p. 14, tradução nossa). A distinção entre as duas abordagens não é bem clara. A abordagem matemática tende a tratar os dados como "determinísticos", além de assumir que as observações não são uma amostra de uma população maior de possíveis observações. Ao invés disso, as observações são geralmente consideradas como a população de interesse. Já os estudos estatísticos tendem a considerar a pontuação particular da força das relações como realizações estocásticas ou probabilísticas de uma verdadeira tendência subjacente ou probabilidade de distribuição de força das relações. Eles também tendem a pensar em um determinado conjunto de dados de rede como uma "amostra" de uma classe ou população maior de tais redes ou elementos e têm a preocupação para que os resultados do presente estudo sejam reproduzidos no "próximo" estudo com amostras semelhantes. Nas seções seguintes, é dada maior ênfase ao lado determinístico, ao invés do estatístico, da análise de redes, pois as redes extraídas são analisadas completamente e não como amostra de uma rede maior.

Este capítulo visa apresentar a análise de redes sociais e suas principais contribuições para a avaliação de indivíduos conforme sua posição social, poder e importância, além de abordar a seleção da amostra, a utilização de métricas na análise e ferramentas que realizam o cálculo das medidas. Ao final, é feito um estudo comparativo entre algumas ferramentas e algumas métricas são selecionadas para integrarem a solução proposta por esse trabalho de acordo com critérios estabelecidos.

2.1 AS ESCOLAS DA ANÁLISE DE REDES SOCIAIS

A análise de redes sociais é uma abordagem oriunda da Sociologia, da Psicologia Social e da Antropologia (FREEMAN, 1996). Segundo Molina (2005), ela é derivada de quatro principais escolas: escola de Manchester, estudos de comunidades, estimação do tamanho das redes pessoais e capital social.

O Quadro 1 apresenta um resumo das principais características das quatro escolas mencionadas.

Escola	Enfoque Teórico	Principais Estudos	Métodos
Escola de Manchester	Complemento do paradigma estrutural - funcionalismo em um mundo urbano fluido.	Barnes (1954); Bott (1955,1957); Epstein (1957, 1963); C. Mitchell (1969); Boissevain (1973); Kapferer (1972)	Sociogramas, observação participante, conceitos sobre teoria de gráficos e álgebra de matrizes.
Estudos de Comunidade	Laços comunitários além dos limites residenciais, apoio social e troca da rede pessoal ao longo do tempo.	Laumann (1973); Fisher (1982); Wellman (1979, 1982, 1988, 1997, 1999); Litwin (1996); Tilburg (1998); Ferrand (1999)	Grandes pesquisas egocêntricas. Bases de dados públicas com dados de redes sociais.
Estimativa do tamanho das redes pessoais	Tamanho, estrutura ou amostras representativas de redes pessoais.	Poole y Kochen (1978); Killworth e Bernard (1978, 1984); Killworth e outros (1998,1990); Freeman e Thompson (1989); Bernard (1990, 1998); McCarty (1997, 2000)	Amostras de listas telefônicas locais ou listas de nomes, “Mundo Pequeno ao contrário, RSW”, método “Scale-up”.
Capital Social	Acesso a pessoas em posições sociais superiores e seus recursos associados.	Lin (1982, 2001); Lin e outros (2001); Burt (1992); Flap e outros (1999); van der Gaag e Snijders (2003)	Gerador de nomes a partir de posições sociais. Gerador de nomes a partir de recursos acessíveis.

Quadro 1 - As quatro principais escolas da análise de redes sociais (Molina, 2005, p. 75, tradução nossa)

“Os antropólogos urbanos da escola de Manchester estavam interessados nas redes sociais para explicar um comportamento que não podia ser explicado por um paradigma teórico estrutural-funcionalista” (MOLINA, 2005, p. 74, tradução nossa). Estes pesquisadores documentaram a relação entre a conduta individual e a estrutura da rede pessoal em situações baseadas em lutas políticas, em clientelismo e em conflitos sociais em locais de trabalho.

A escola de estudos de comunidade surgiu em função da preocupação com as mudanças no estilo de vida tradicional. Essa escola se tornou conhecida pela pesquisa focada na localização das redes de apoio social, constituídas principalmente por parentes, amigos e vizinhos que proporcionam informação, socialização e ajuda em geral (MOLINA, 2005, p.

78). Este estudo, iniciado nos anos setenta e que se prolongou até meados dos anos noventa, criou uma grande base empírica para formular as características globais das redes pessoais na sociedade americana e canadense.

O estudo de estimação do tamanho das redes pessoais se inicia na busca pela resposta ao questionamento “Quantas pessoas você conhece?”. A resposta a essa pergunta dá lugar ao estudo que conclui que não existe uma resposta única para essa questão. Nas pesquisas foram realizados experimentos que realizavam o cálculo do volume total de conhecidos de uma pessoa utilizando listas telefônicas e a aplicaram critérios como contatos acumulados, contatos ativos e laços fortes para auxiliar na definição do tamanho das redes pessoais.

Já o estudo do capital social está centrado em três grupos: o capital social com foco na pessoa, o capital social centrado na rede e o capital social focado na rede de associações (sociedades civis). No primeiro caso, o capital social é visto como algo inerente às pessoas, ou seja, ao número e a qualidade de relações, determinado por sua classe social. Para o segundo grupo, os recursos do capital social residem nas propriedades da rede de relacionamento, mais do que nas pessoas. Por fim, o terceiro conceito de capital social associa o êxito econômico de uma região do país com a rede de entidades civis e econômicas existentes nela.

2.2 APLICAÇÕES E BENEFÍCIOS DA ANÁLISE DE REDES SOCIAIS

Para Perer (2006), a análise de redes sociais surgiu como um poderoso método para compreender a importância das relações entre as pessoas. Abaixo, seguem alguns exemplos de aplicações da análise de redes sociais:

- Na área médica, a análise de redes sociais é aplicada para estudar a propagação e evolução de diversos tipos de doenças. Cohen e outros (2000), por exemplo, realizou um estudo sobre como uma rede social diversa influencia na saúde do indivíduo.

- No campo da sociologia, a análise de redes sociais é utilizada para estudar a formação das redes sociais humanas, onde pesquisas mostram que existe uma tendência de aproximação das pessoas com interesses mútuos.

- Na área de comunicação, o estudo das redes sociais pode ser utilizado para identificar as dificuldades de comunicação entre pessoas de um mesmo processo chave ou grupo no interior de uma organização. Pesquisas mostram que a dificuldade de comunicação surge, na

maioria das vezes, devido à grande fragmentação existente ao longo do processo (fronteiras funcionais, hierárquicas e físicas).

- Em tecnologia, a análise de redes sociais apresenta-se como uma ferramenta útil no campo da Gestão de Conhecimento (BUSCH e outros, 2001; CROSS e outros, 2001; PARKER e outros, 2001) e também em sistemas de recomendação de especialistas, por exemplo. Nesses sistemas, pode-se aplicar o estudo das redes sociais em dois momentos: na utilização da opinião das pessoas da rede para indicar o grau de competência de uma determinada pessoa ou na ordenação de uma lista de especialistas priorizando os mais próximos da rede social da pessoa que fez a busca. Em Mcdonald (2003), a técnica é utilizada para filtrar e ordenar, de acordo com a proximidade na rede social do usuário que fez a pesquisa, a resposta à busca de um especialista.

As diferenças na forma como os indivíduos se conectam podem ser extremamente importantes para compreender os seus atributos e comportamentos. Os indivíduos com maior número de relacionamentos muitas vezes estão expostos a maior quantidade e variedade de informação. Desta forma, indivíduos altamente conectados podem ser mais influentes ou podem ser mais influenciados pelos outros.

A forma como a população inteira está interligada também pode influenciar bastante. Doenças e rumores, assim como informação útil, se espalham de forma mais rápida onde as taxas de ligação são maiores. Populações com maior densidade de conexões geralmente possuem maior quantidade e diversidade de perspectivas para auxiliar a resolução de problemas.

Como a maioria dos indivíduos não está normalmente ligada diretamente com a maioria dos outros indivíduos da população, pode ser muito importante analisar além das conexões imediatas dos atores. O estudo, para ser bem feito, geralmente deve levar em consideração medidas relacionadas com a distância entre os agentes. Alguns atores podem ser capazes de chegar à maior parte dos outros membros da população com pouco esforço. Na prática, ao passar informação para seus amigos, eles diriam aos amigos deles e rapidamente todos, ou quase todos, já teriam a informação. Outros podem ter dificuldade, pois mesmo informando as pessoas aos quais se relacionam, elas não estão bem relacionadas e a mensagem não consegue ir longe. Desta forma, se os amigos de uma pessoa têm um ao outro como amigo, ela tem uma rede limitada, mas, se os amigos dela têm muitas conexões não-

sobrepostas, seu leque de ligações aumenta. Com efeito, uma grande diferença entre "classes sociais" não se dá tanto com relação ao número de conexões que os atores possuem, mas em saber se estes relacionamentos sobrepõem e constrangem ou estendem e oferecem oportunidades. Diferenças na forma como a rede é interligada auxiliam na a compreensão das diferenças nas macro-propriedades dos grupos sociais como difusão, homogeneidade, solidariedade e outras.

Alguns benefícios esperados da aplicação da análise de redes sociais observados por (CROSS; PARKER, 2004) são os seguintes: integrar a rede de pessoas que participam de processos de negócios em uma empresa, identificar indivíduos centralizadores de informação da rede pesquisada, motivá-los para disseminar informações entre seus colegas e possibilitar a avaliação do desempenho de um grupo de pessoas que deve trabalhar de forma integrada.

Para Hanneman (2001, p. 40, tradução nossa), “as diferenças na forma como os atores estão conectados em uma população podem ser um indicador chave da solidariedade, ‘densidade moral’, e ‘complexidade’ da organização social da população”. Ainda segundo ele, “o número e os tipos de vínculos que os atores têm são chaves para determinar quanto o seu enraizamento na rede restringe o seu comportamento e o leque de oportunidades, influência e o poder que eles têm” (HANNEMAN, 2001, p. 40, tradução nossa).

2.3 SELEÇÃO DA AMOSTRA

O objetivo da análise de redes sociais é demonstrar que a análise de uma um par de atores relacionados só tem sentido em relação ao conjunto das outros pares de atores relacionados na rede, porque sua posição estrutural tem necessariamente um efeito sobre sua forma, seu conteúdo e sua função (MARTELETO, 2001, p. 72). Isso quer dizer que a análise de redes sociais pressupõe um raciocínio diferente da análise tradicional de informações. Ao invés de pensar em como as ligações entre um ator e outro descrevem os atributos do “ego”, analistas de redes sociais vêem a estrutura de ligações no âmbito da qual o ator está embutido. Atores são descritos por suas relações e não por seus atributos e as relações propriamente ditas são tão importantes quanto os atores que elas conectam. Isto significa que os atores não são normalmente escolhidos de modo independente, tal como em muitos outros tipos de estudos.

Considerando um estudo de laços de amizade, após selecionar uma pessoa para a amostra, ela é questionada para que identifique uma quantidade de amigos. A partir daí, é

necessário rastrear cada um desses amigos para interrogá-los a respeito de seus laços de amizade. Essas pessoas estarão na amostra porque a primeira pessoa estava e, portanto, esses elementos não são "independentes". Desta forma, ao invés de utilizar uma amostragem probabilística independente, como nos estudos tradicionais, o estudo de redes tende a incluir todos os atores que ocorrem dentro de uma ou mais fronteiras. Naturalmente, a população incluída no estudo de uma rede pode ser uma amostra de uma população maior. A utilização de populações inteiras, como forma de seleção faz com que seja importante que o analista esteja consciente das fronteiras de cada população a ser estudada.

2.4 ANÁLISE DE REDES SOCIAIS A PARTIR DE MÉTRICAS

Construir representações visuais de redes sociais tem provido os analistas com visões sobre a estrutura da rede, assim como tem sido um auxílio para a comunicação de fenômenos na rede (FREEMAN, 2000). Apesar de esta ser uma forma muito comum de análise de redes sociais, ela traz em si a necessidade de preparação e conhecimento prévio por parte da pessoa que está analisando o gráfico. Interpretar essas visualizações é desafiador porque é difícil compreender as características e a estrutura das redes quando há muitas arestas e nós, e, também, os sistemas atuais são, muitas vezes, uma mistura de métodos estatísticos e uma saída visual poluída que deixa muitos analistas incertos quanto à maneira de explorar o grafo de uma forma ordenada (PERER, 2006, p. 1). Ou seja, não adianta entregar um grafo com uma rede social para um leigo olhar que ele não irá tirar muitas conclusões em função disso.

Além disso, muitos trabalhos relacionados a isso já foram publicados e muitas ferramentas foram construídas. Moody, McFarland e Bender-deMoll (2005) desenvolveram um sistema chamado Sonia Network Image Animator para visualização de redes sociais dinâmicas, ou seja, levando em consideração quando as relações entre os nós ocorreram. Huffaker e outros (2008) apresentaram o Otter, uma ferramenta para visualização dados arbitrários de uma rede. Nela, o usuário pode interagir utilizando cores, movimentos, exibição de atributos e utilização de zoom para melhorar a forma como os dados estão exibidos e dispostos. Outra ferramenta para visualização de redes sociais é a KrackPlot (KRACKPLOT, 2008). Através dela pode-se visualizar diretamente grafos gerados pela ferramenta UCINet (BORGATTI; EVERETT; FREEMAN, 2002), que é analisada na seção 2.4.1. Além desses trabalhos, existem muitas outras ferramentas disponíveis como o yEd (YED, 2008) e os *browsers* da TouchGraph (TOUCHGRAPH, 2008). Mantido pela yWorks, o yEd é um editor de grafos que permite, entre outras coisas, gerar automaticamente o layout de grafos com

estruturas complexas. Já o TouchGraph Google browser e o TouchGraph FaceBook Browser permitem explorar as conexões entre sites relacionados e entre os usuários no Facebook, respectivamente.

A utilização de métricas, assim como a visualização da rede tem sido abordada em inúmeros trabalhos e pesquisas. Entretanto, dois pontos foram cruciais na escolha da análise da rede social através de métricas, ao invés de visualização (implementar a visualização como parte da solução de análise da rede caracteriza-se como um projeto futuro para ampliação do escopo deste trabalho). O primeiro motivo é justamente o fato de uma medida trazer sempre um resultado numérico, o que reduz a necessidade de conhecimento do analisador das informações. Assim, com apenas algumas instruções, um leigo já consegue avaliar alguma coisa sobre os integrantes da rede de posse dos resultados de cálculos de medidas. O outro incentivo se deu em função de sempre estarem surgindo novas medidas. Assim, preparar uma ferramenta para novas métricas que estão surgindo ou irão surgir apresentou-se como uma oportunidade interessante.

A seção 2.4.1 apresenta algumas das ferramentas de cálculo de métricas aplicadas a redes sociais existentes no mercado e a seção 2.4.2. realiza um comparativo entre elas.

2.4.1 Ferramentas de análise de redes sociais através de medidas

Muitas são as ferramentas existentes no mercado que realizam o cálculo de métricas em redes sociais. Algumas das principais delas, por serem mais conhecidas e utilizadas, foram selecionadas e serão abordadas nesta seção.

Dentre as ferramentas analisadas, duas delas são ferramentas disponíveis na *web*. A primeira delas é a Tyna (TYNA, 2008): um sistema para gerenciamento, comparação e mineração de múltiplas redes, tanto direcionadas, como não. Eficientemente, tYNA implementa métodos que provaram ser úteis em análise de redes, incluindo o cálculo de estatísticas globais e a detecção de *hubs* e gargalos.

A ferramenta tYNA também permite gerir um grande número de redes privadas e públicas através de um sistema de marcação flexível, filtrá-las baseado em uma variedade de critérios e visualizá-los através de uma interface gráfica interativa. “Uma série de dados biológicos comumente utilizados foram pré-carregados na tYNA, padronizados e agrupados em categorias diferentes” (YIP e outros, 2006, p. 1, tradução nossa).

Comparando com as demais ferramentas, tYNA apresenta uma quantidade menor de métricas, o que dificulta um pouco uma análise mais detalhada através dela. Entretanto, seu diferencial se dá na manipulação de mais de uma rede, gerando ao final uma rede resultante que pode ser a união, interseção ou a diferença entre elas.

A Figura 1 mostra a interface de visualização e análise da tYNA. Nela, o usuário pode fazer operações de zoom e movimentação do gráfico como um todo, além de utilizar cores para destacar os nós de acordo com suas propriedades. Abaixo, pode ser vista uma tabela com as estatísticas gerais da rede. Os resultados das estatísticas individuais podem ser vistos fazendo o *download* de um arquivo texto com os valores através de um link na tela. A Figura 1 também mostra o arquivo texto para a rede analisada.

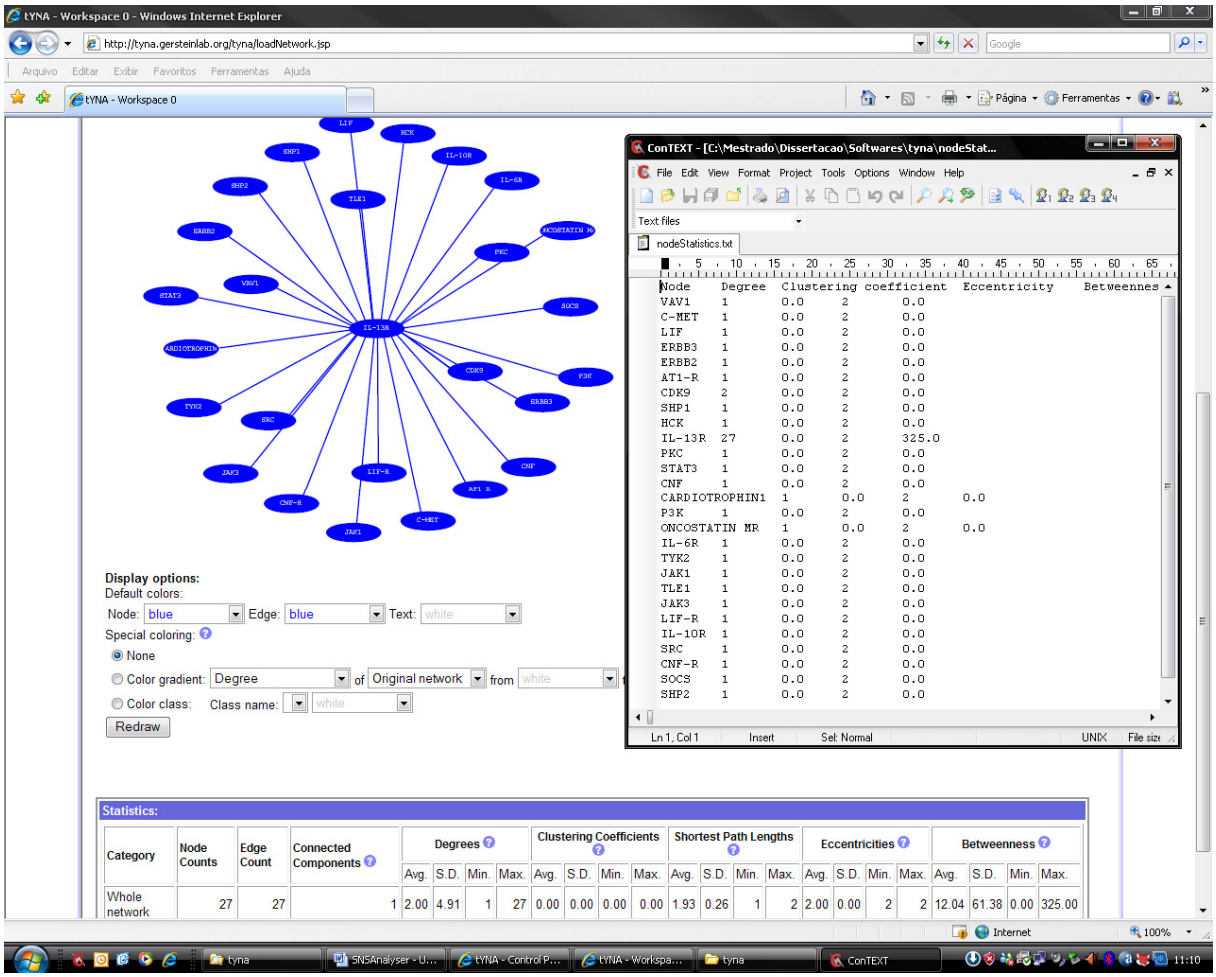


Figura 1 - tYNA – Interfaces visualização da rede e das estatísticas individuais

A outra ferramenta *on-line* analisada foi o Netvis (NETVIS, 2008). Desenvolvido por Jonathon N. Cummings, ela se caracteriza como uma ferramenta disponível na *web* de código aberto destinada a pesquisadores para simular, analisar e visualizar redes sociais, utilizando

dados de pesquisas disponíveis na Internet, importação de arquivos com valores separados por vírgula, grupos de discussão eletrônicos ou dados de equipes de trabalho geograficamente separadas.

O NetVis tem como ponto forte o fato de possibilitar a visualização gráfica das redes analisadas. Entre os seus principais pontos fracos está a pequena quantidade de métricas calculadas, algumas delas, inclusive apenas para a rede como um todo e não para cada indivíduo.

A Figura 2 mostra duas interfaces do NetVis, uma sobre a outra, onde se pode ver a edição da rede em forma de tabela e a visualização de seu grafo. A pequena tabela no topo da imagem apresenta o resultado dos cálculos das medidas da rede como um todo e *links* para as medidas calculadas para cada nó.

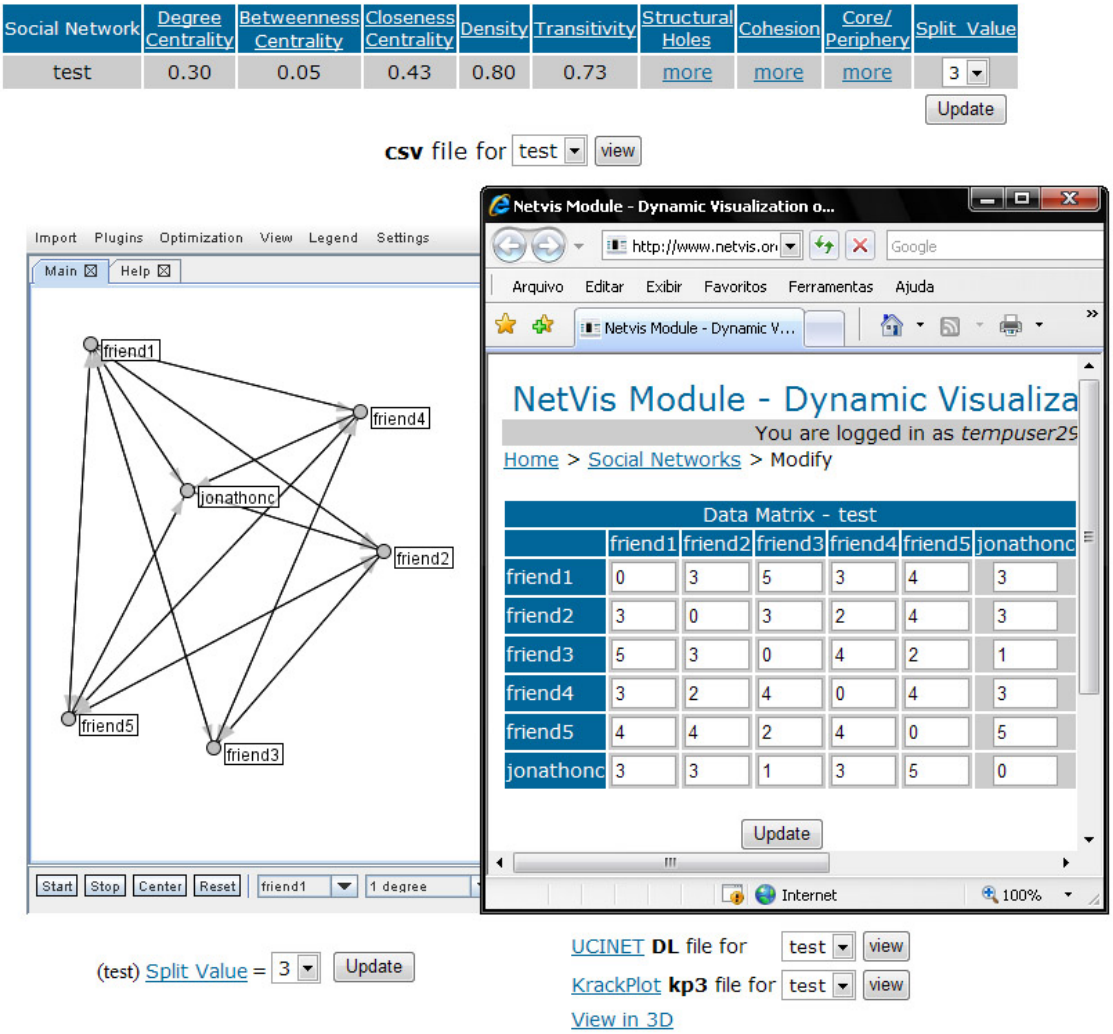


Figura 2 - NetVis – Interfaces para edição e análise da rede social

Uma das ferramentas pesquisada foi a Agna, uma ferramenta projetada para realizar análise de redes sociais, sociometria e análise seqüencial (AGNA, 2008). Ela pode auxiliar no estudo das relações da comunicação em grupo, análise organizacional, construção de equipes, relações de parentesco ou familiares e comportamento animal (BENTA, 2003).

Comparando com outras aplicações, Agna é uma ferramenta amigável e fácil de aprender que possibilita o aprofundamento da filosofia e dos conceitos por trás da análise de redes sociais. A possibilidade de criar, editar, armazenar, visualizar, importar e exportar redes sociais contribui para que ela seja avaliada como uma boa ferramenta, apesar de outras ferramentas terem ainda mais recursos e métricas que ela. Abaixo, a Figura 3 retrata a interface disponibilizada pela Agna para a análise de redes sociais.

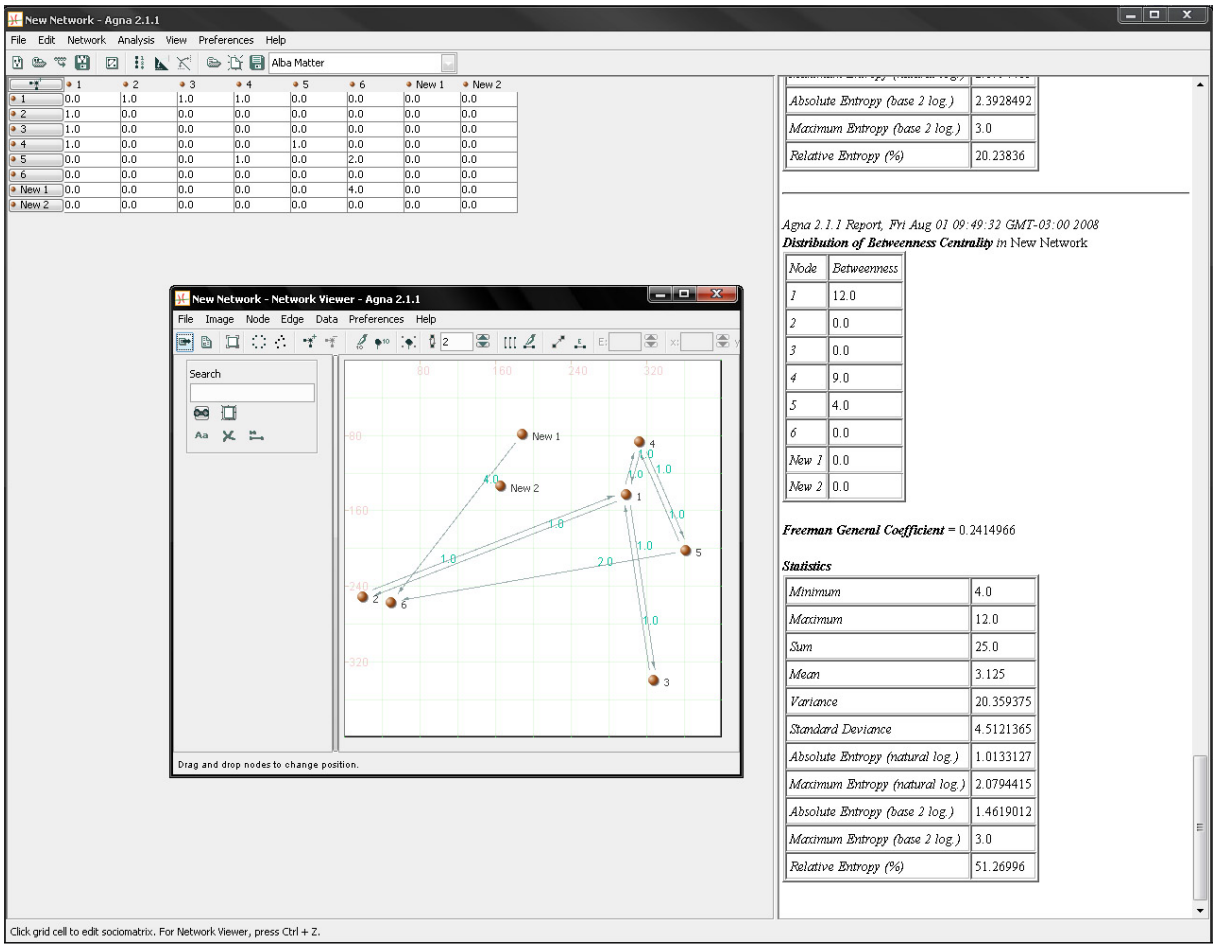


Figura 3 - Agna – Interface para análise de redes sociais

O NetMiner (NETMINER, 2008) é um produto da empresa Cyram para realizar análise geral de redes. Ele possibilita a exploração da sua rede de dados de forma visual e interativa e ajuda a detectar padrões subjacentes e estruturas da rede. Ele inclui muitas rotinas

de análise, incluindo interação entre pares, modelo de influência de rede social e coesão de blocos.

Dentre as ferramentas analisadas, o NetMiner se apresentou como uma das opções mais completas para análise de redes de um modo geral (sem estatística avançada). Uma gama enorme de métricas e operações, a exibição e edição da rede em forma de tabelas ou gráfico, a exibição de resultados numéricos em tabelas e visualmente destacados em gráficos, junto com a opção de importação e exportação em diversos formatos ajudaram bastante para classificar assim essa ferramenta.

O NetMiner foi concebido considerando a ação do usuário para a análise, além da análise propriamente dita e do desempenho dos algoritmos. A Figura 4 mostra como estão dispostos os objetos de interação do usuário nessa ferramenta.

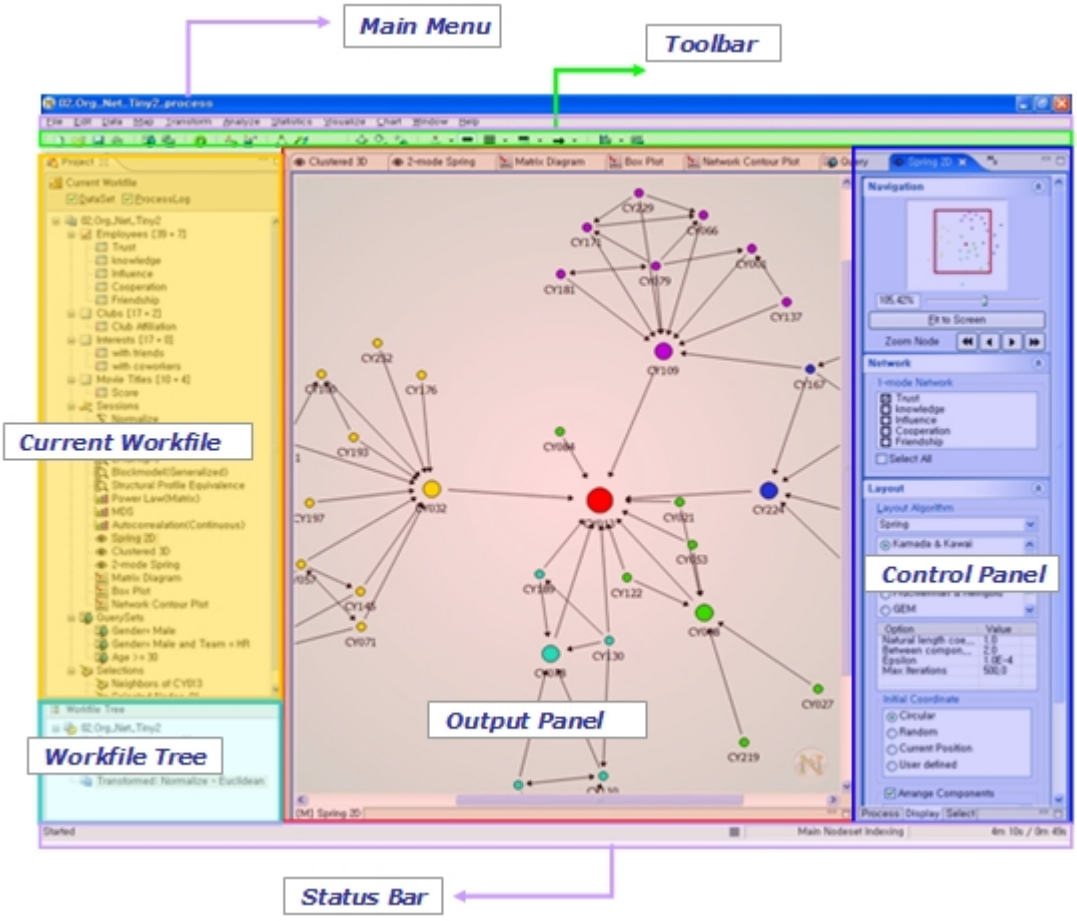


Figura 4 - NetMiner – Disposição dos objetos na tela

Já o StOCNET (STOCNET, 2008), é um software aberto para a análise estatística de redes sociais utilizando modelos estatísticos avançados. Ele fornece uma plataforma para

tornar disponível para um público mais vasto uma série de métodos estatísticos, que anteriormente eram propriedade privada, apresentada em módulos separados, e permite que novas rotinas sejam facilmente implementadas (HUISMAN; VAN DUIJN, 2003, p. 8).

Desenvolvido em colaboração com engenheiros de software da Science Plus (SCIENCE, 2008) e pesquisadores, o StOCNET tem como principais recursos a execução de métodos estatísticos (estocásticos) diferentes, o cálculo de algumas estatísticas de rede descritivas comuns, transformação e/ou seleção de dados e simulação de possibilidades. Comparando com outras ferramentas, o diferencial dos métodos incluídos no StOCNET é que eles são baseados em modelos de probabilidade explícitos para redes.

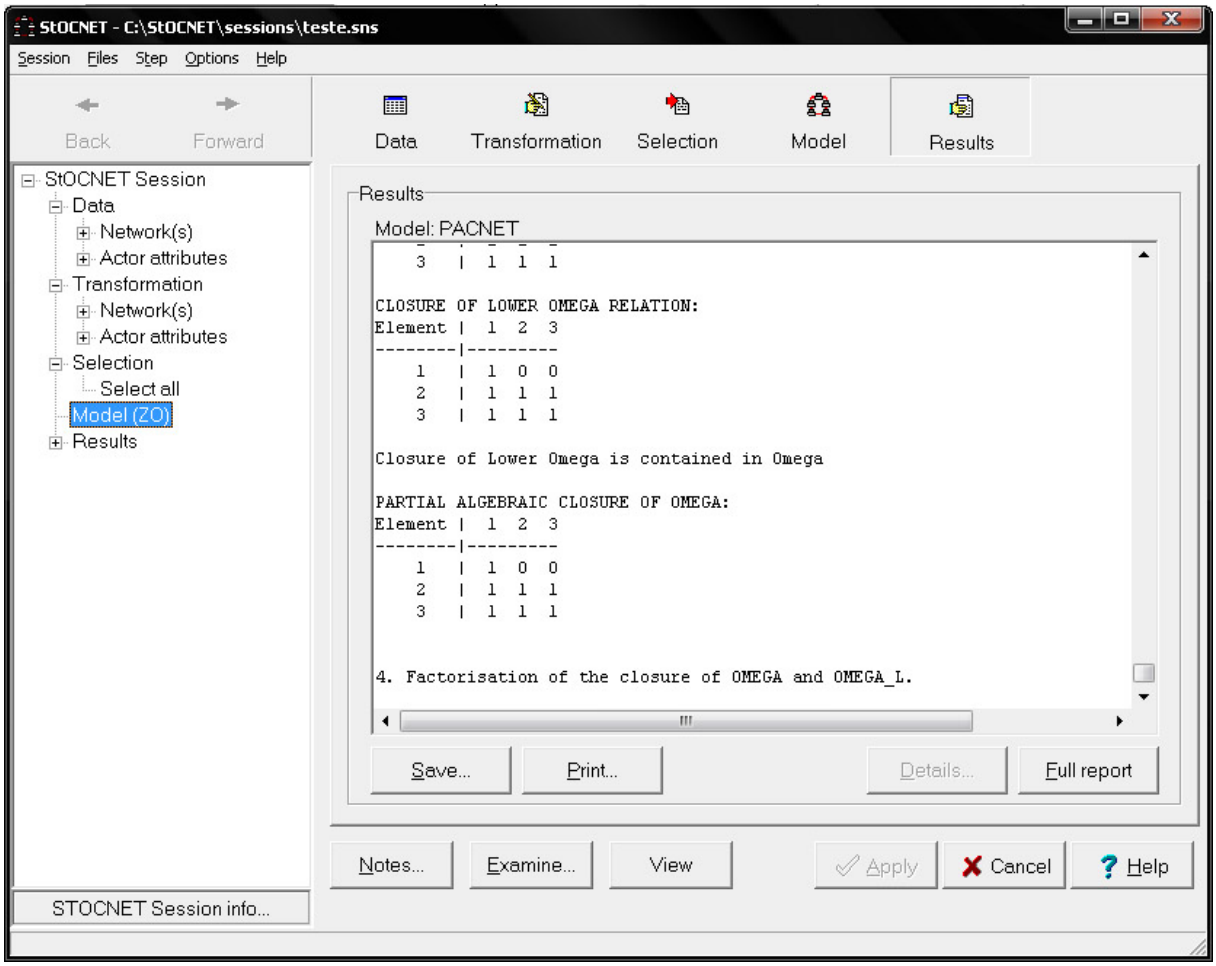


Figura 5 - StOCNET – Interface de exibição de resultados

O Pajek (PAJEK, 2008) é um programa para análise de grandes redes. Com seu desenvolvimento iniciado em 1996, o Pajek teve como principal motivação a observação de que já existiam informações que podiam originar grandes redes armazenadas em formatos

interpretáveis por máquinas. O Pajek deveria então se colocar como uma ferramenta para análise e visualização dessas grandes redes.

Redes de colaboração, de Internet, de citação, de difusão (doenças, notícias, inovações) e outras cujo tamanho passa dos milhares ou milhões de nós não conseguem ser manipuladas de forma eficiente utilizando as ferramentas de análise de redes genéricas, uma vez que elas se baseiam no modelo de representação dos dados através de matriz. Para atender à demanda de análise desse tipo de rede é que o Pajek foi projetado. A Figura 6 mostra as interfaces do Pajek, destacando a interface principal do sistema acima, o relatório de resultados no centro e a interface visual abaixo.

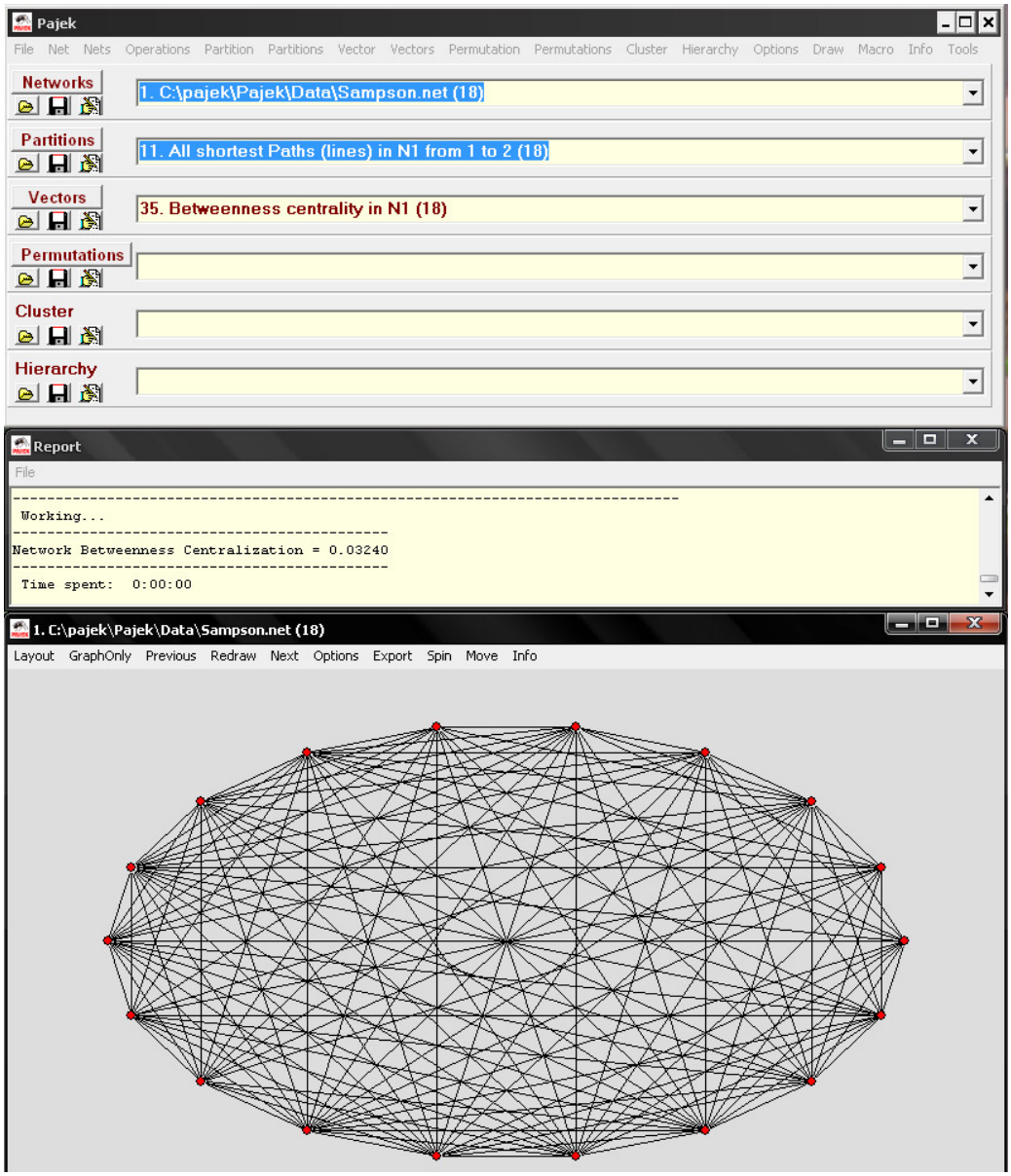


Figura 6 - Pajek – Principais telas do sistema

Durante a avaliação, pode-se perceber que o Pajek tem menos procedimentos estatísticos que outros como o UCINet (BORGATTI; EVERETT; FREEMAN, 2002), o NetMiner (NETMINER, 2008) e o StOCNET (STOCNET, 2008), mas seu posicionamento com foco em grandes redes e suas opções avançadas de visualização fazem com que muitas ferramentas exportam sua rede para um formato reconhecido pelo Pajek.

O UCINet (BORGATTI; EVERETT; FREEMAN, 2002) é um programa abrangente para a análise de redes sociais e outros dados proximidade. Ele é provavelmente o pacote de software mais conhecido e mais frequentemente utilizado para análise de dados de redes sociais, pois contém inúmeras rotinas de análise de redes.

O UCINet lê e escreve arquivos em uma quantidade variada de formatos e é distribuído em conjunto com outros sistemas como o Pajek, o Maje e o NetDraw para suprir a inexistência de procedimentos de visualização de redes desse sistema. Apesar de se limitar a trabalhar com redes que possuem no máximo 32.767 vértices, na prática, muitos procedimentos ficam bastante lentos quando aplicados a redes com um número entre 5.000 e 10.000 nós. A Figura 7 mostra a tela principal do sistema e dois logs de saída, um de importação de dados e o outro de exibição de resultados de cálculo.

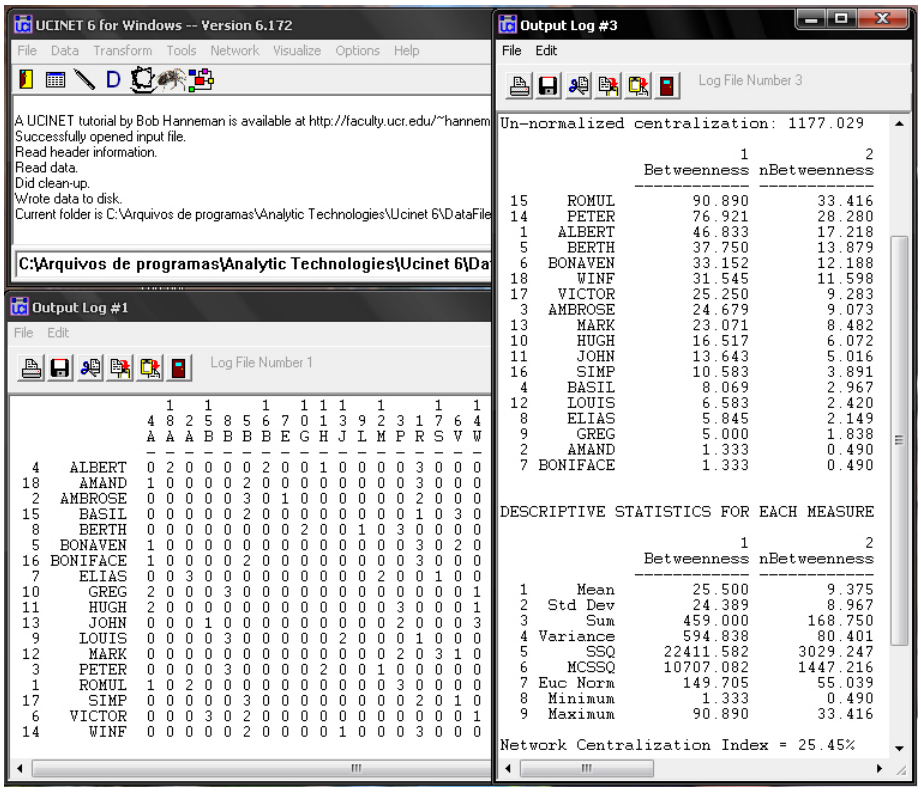


Figura 7 - UCINet – Tela principal e Logs de saída

Assim como o NetMiner, o UCINet mostrou-se como uma das opções mais completas para análise de redes de um modo geral (sem estatística avançada, como é o caso do StOCNET). A ausência de um módulo de visualização, garantindo a exibição gráfica da rede e dos resultados dos cálculos é um ponto fraco remediado pelas outras ferramentas incluídas em seu pacote. Por outro lado, a ampla abrangência de métricas, procedimentos e operações auxiliados pela grande variedade de opções de formatos de importação e exportação de arquivo eleva bastante o conceito da ferramenta.

2.4.2 Comparativo entre ferramentas de análise de redes sociais

Conforme visto na seção 2.4.1., em função da necessidade eminente de mecanismos que possibilitem a análise de redes sociais, inúmeras ferramentas foram desenvolvidas, sob diferentes modelos de arquitetura e com distintas propostas tecnológicas. A fim de proporcionar aos usuários uma visão comparativa entre elas, se faz necessária a definição de categorias para a classificação das ferramentas de análise de redes sociais.

Com base na revisão de literatura e no tipo de análise que interessa neste trabalho, foram definidos alguns critérios para categorização das ferramentas de análise de redes sociais, conforme listado a seguir:

- Objetivo Principal – representação da funcionalidade essencial da ferramenta, ou seja, o objetivo principal para o qual o *software* foi construído.
- Métricas Calculadas – indicação da abrangência de métricas utilizadas na ferramenta. A maior quantidade de métricas indica uma ferramenta mais completa. Entretanto a ausência de alguma métrica importante também pode contar pontos negativos para a ferramenta em questão.
- Interface com outros sistemas – definição das interfaces de importação e exportação de dados para utilização em outras ferramentas.
- Forma de exibição da rede e dos resultados – descrição da maneira como a rede é exibida e manipulada e os resultados do processo de calculo são exibidos para o usuário.

O Quadro 2 retrata o enquadramento das ferramentas de busca de análise de redes sociais descritas no presente trabalho nos critérios supra relacionados.

Ferramenta	Objetivo Principal	Métricas Calculadas	Interface com Outros Sistemas	Forma de Exibição da Rede e dos Resultados
tYNA	- Permitir o gerenciamento, comparação e mineração de múltiplas redes, tanto direcionadas, como não.	- Calcula a quantidade total de nós e conexões e componentes conectados na rede. - Calcula a média, valor mínimo, máximo e desvio padrão para a rede como um todo em relação às seguintes métricas: Degrees, Clustering Coefficients, Shortest Path Lengths, Eccentricities e Betweenness. - Calcula para cada nó: Degree, Clustering coefficient, Eccentricity e Betweenness.	- Importa redes de arquivos de texto em três formatos: Arquivos com valores separados por vírgula, separados por tab e Simple Interaction File (.sif) - Importa categorias em arquivos texto do tipo Simple category File (.sif) - Exporta os resultados de estatísticas individuais para arquivo texto. - Exporta o gráfico da rede para: BMP, GDL, PDF, PNG, OS, RAW e SVG. - Disponibiliza uma inteface através de Web Service.	- Exibe a rede através da visualização do grafo. - Os resultados dos cálculos para a rede como um todo são exibidos em tabelas na própria tela do browser. - Os resultados das estatísticas individuais são disponibilizados através de link para o arquivo de texto que contém os resultados.
NetVis	- Dar a oportunidade de explorar redes sociais através de uma aplicação online, que integra a análise de dados e a visualização.	- Calcula a Degree Centrality, Betweenness Centrality, Closeness Centrality, Density e Transitivity para a rede como um todo, além dos Structural Holes (Constraint, Effective size, Efficiency e Hierarchy), Cohesion (Shortest Path, Reachability e Geodesics) e Core/Periphery de cada nó.	- Importa dados de arquivos de texto em três formatos: csv, lista de nós e eventos. - Exporta a rede para arquivos csv, datasets do UCINet (.dl) e arquivos do KrackPlot (.kp3) - Exporta os resultados de alguns cálculos para arquivos csv.	- Exibe a rede em forma de tabela editável e através da visualização do grafo. - Os resultados dos cálculos são exibidos em tabelas que podem ser copiadas e exportadas para arquivos csv em alguns casos.
Agna	- Auxiliar na análise de redes sociais, sociometria e análises seqüenciais.	- Exibe informações gerais (uma descrição básica, alguns parâmetros estruturais: Diâmetro, densidade, Coesão) - Informações sobre os menores caminhos (matriz geodésica, menores caminhos) - Coeficientes no nível de nó (quatro tipos de coeficientes de centralidade e alguns coeficientes sociométricos).	- Importa e exporta redes em: Arquivos com valores separados por vírgula (.csv) e separados por tab (.txt, .dat e .text) - Exporta Visualização das Redes em: Arquivos JPEG (.jpg).	- Exibe a rede em forma de tabela editável e através da visualização do grafo também editável. - Os resultados dos cálculos são exibidos em tabelas no formato HTML e podem ser salvos.
NetMiner	- Possibilitar a análise e visualização exploratória de dados de redes.	- Análise de vizinhança (Degree, Ego Network, Structural Hole, ...), de subgrafos (Dyad Census, Tryad Census, ...), de conexões (Shortest Path, Dependency, Conectivity, ...), coesão (Clique. k-Plex, Lambda Set, ...), centralidade (Closeness, Betweenness, Power, ...), equivalência, posição, modelos e propriedades	- Importa e exporta redes em: Arquivos de texto com qualquer separador (.txt, .csv), Excel (.xls), NTF (.ntf), DL (.dl, .dat, .txt), Pajek (.net, .vec, .clu, .per), Stocnet (.dat, .txt) - Importa redes também em: arquivos GML.	- Exibe a rede em forma de tabela editável e através da visualização do grafo também editável. - Os resultados dos cálculos são exibidos em tabelas que podem ser copiadas e em gráficos onde pode ocorrer interação com usuário.

Stocnet	- Possibilitar a análise estatística de redes sociais utilizando modelos estatísticos avançados.	- Na categoria “Descrições Estatísticas”: grau variância e métodos diádicos e triádicos estão disponíveis. - Na categoria “Comparação e Classificação de Atores”: modelagem de blocos estocástica, construção de álgebras parciais e estimação de estruturas transitivas latentes.	- Exporta a rede para: Multinet, Pajek, Netminer e Structure. - Importa de arquivos de texto separados por espaço. - A ferramenta inclui interface para 5 programas de modelagem estatística de redes sociais: SIENA, BLOCKS, P2, ULTRAS e ZO.	- Exibe a rede em forma de tabela. - Os resultados dos cálculos, em sua maioria, são exibidos em relatório com tabelas que pode ser salvo.
Pajek	- Apoiar a abstração de grandes redes através de sua decomposição em redes menores e oferecer ao usuário uma seleção eficiente de algoritmos para a análise de grandes redes e uma ferramenta poderosa de visualização.	- Calcula medidas de centralidade (closeness e betweenness) para a rede como um todo, encontra vértices importantes, buracos estruturais, coeficientes de cluster e outras medidas que podem ser aplicadas a partes menores da rede.	- Importa e exporta a rede em: Vega graphs(.vgr), Ball and Stick (.bs), Ucinet DL (.dat), MDL MOL (.mol). - Importa a rede em: Gedcom (.ged), Mac Molecule (.mac). - Exporta a rede em: GraphML (.xml). - Exporta a visualização da rede em: EPS/PS (.eps, .ps), Scalable Vector Graphics (.svg), Bitmap (.bmp), X3D (.x3d), Kineimages (.kin), VRML (.wrl).	- Exibe e permite edição da rede egocêntrica de um determinado nó. - Os resultados são exibidos em um relatório que pode ser salvo como arquivo texto.
UCINet	- Possibilitar uma análise abrangente de redes sociais assim como dados 1-mode e 2-mode.	- Calcula medidas de centralidade, identificação de subgrupos, análise de papéis, teoria dos grafos elementar e análise estatística baseada em permutação. Além disso, o pacote tem fortes rotinas de análise de matrizes.	- Importa e exporta redes em: DL, KrackPlot, Pajek, RAW e Excel - Importa redes em: VNA, Negopy e Arquivo texto. - Exporta redes em: Mage e Metis.	- Exibe a rede em forma de tabela editável. - Apesar de não conter procedimentos para visualização das redes, é distribuído em conjunto com as ferramentas Mage, NetDraw, e Pajek, para que as redes sejam visualizadas através delas. - Os resultados são exibidos em um relatório que pode ser salvo como arquivo texto.

Quadro 2 - Comparativo entre ferramentas de análise de redes sociais

A partir da categorização definida no Quadro 2 é possível observar as principais características e funcionalidades de cada ferramenta, bem como analisar semelhanças e diferenças existentes entre as mesmas.

Dentre as ferramentas avaliadas, as que possem menos recursos são as disponíveis na *web*: tYNA e NetVis. Entretanto, isso não justifica o fato de serem menos poderosas. A tYNA permite a visualização da rede, mas a interação do usuário com o grafo é pequena. Já o NetVis apresenta uma visualização da rede mais interativa. Por sua vez, a tYNA oferece a opção de

manipulação de mais de uma rede, realizando operações entre ela, o que o NetVis não faz. Comparando com as demais, ambas apresentam uma quantidade menor de métricas.

Como já foi dito anteriormente, a Agna é uma ferramenta amigável e de fácil aprendizado. Mais evoluído que os sistemas anteriores, ele possibilita criar, editar, armazenar, visualizar, importar e exportar redes sociais com facilidade. No que tange ao cálculo de métricas, entretanto, ele está numericamente bem aquém das ferramentas NetMiner, StOCNET, Pajek e UCINet, abordadas a seguir.

O Pajek tem um foco em grandes redes e uma quantidade grande de opções avançadas de visualização que faz com que ele tenha lugar de destaque na análise de redes sociais.

O StOCNET é uma das ferramentas, juntamente com o NetMiner e o UCINet, que possui maior quantidade de procedimentos estatísticos e maior variedade de tipos de procedimentos de análise. Contudo, o StOCNET é um sistema mais especializado e foi projetado especialmente para realizar análises estatísticas avançadas.

Por fim, as melhores ferramentas para realizar uma análise abrangente de redes sociais através de métricas foram o NetMiner e o UCINet. O NetMiner apresenta um poder maior no que tange à visualização das redes e dos resultados. O UCINet utiliza outras ferramentas incluídas em seu pacote, para possibilitar a visualização. Todavia, ambas incluem um amplo leque de métricas, estatísticas, operações e métodos de análise baseados em procedimentos. Além disso, apresentam uma enorme variedade de formatos de importação e exportação de arquivos.

2.5 SOLUÇÃO PROPOSTA

Em função da grande quantidade de ferramentas para análise de redes sociais existentes e também da excelente qualidade apresentada por algumas delas, não é interesse deste trabalho construir uma ferramenta onde o módulo de análise fosse tão abrangente como algumas ferramentas já estabelecidas como o NetMiner e o UCINet. Entretanto a solução proposta deve prever a possibilidade de análise das redes sociais para que ela se torne completa e independente de outros sistemas.

Assim, a melhor opção é incluir na ferramenta construída um módulo de análise com métricas selecionadas de acordo com um critério que possibilite a análise da importância de cada membro na rede em questão.

Primeiramente, faz-se necessário que não sejam calculadas métricas para a rede como um todo, mas apenas as que resultem em números para cada indivíduo, uma vez que o objetivo é analisar como cada um se posiciona na rede e selecionar os que são mais interessantes para o pesquisador.

Deve-se optar também por medidas que indiquem se os indivíduos têm poder e influência na rede em que se encontram. Segundo Hanneman (2001, p.75, tradução nossa), “o poder surge a partir da ocupação de posições vantajosas em redes de relacionamento. Três fontes básicas de obtenção de vantagem são alto grau de centralidade, alto grau de proximidade e alto grau de intermediação”. Desta forma, sete métricas relacionadas a isso são implementadas na ferramenta construída ao longo deste trabalho. Três estão ligadas ao grau de centralidade - o grau de centralidade propriamente dito (*degree*), o grau de centralidade normalizado (*normed degree*) e a participação relativa ao grau da rede (*share*). Duas estão relacionadas ao grau de proximidade - o grau de distanciamento (*farness*) e sua normalização, o grau de proximidade (*closeness*). E duas são associadas ao grau de intermediação – o grau de intermediação propriamente dito (*betweenness*) e o grau de intermediação normalizado (*normed betweenness*).

Assim, permite-se que o pesquisador faça uma análise a respeito da centralidade, além de inferir a respeito da importância e poder de cada um dos avaliados na rede. As métricas foram obtidas durante a revisão da literatura e também através da documentação de algumas ferramentas analisadas. O Quadro 3 é uma síntese das informações coletadas onde se descreve como as métricas selecionadas podem ser obtidas.

Métrica	Descrição
Degree	Calcula a quantidade de vértices adjacentes a um determinado nó da rede.
Normed Degree	É o grau (degree) dividido pelo maior grau (degree) possível, multiplicado por 100 (expresso em percentual).
Share	É o grau (degree) do nó dividido pela soma dos graus (degrees) de todos os nós.
Farness	É a soma dos tamanhos dos caminhos de um vértice para cada outro nó da rede.
Closeness	É o menor farness possível dividido pelo farness do nó, multiplicado por 100 (expresso em percentual).
Betweenness	É a soma das proporções de vezes que um nó aparece no menor caminho entre dois pares de vértices distintos.
Normed Betweenness	É o betweenness dividido pelo maior betweenness possível, multiplicado por 100 (expresso em percentual). Pode ser obtido através da fórmula: 2*betweenness/(n^2–3*n+2)*100, onde n = nº de nós.

Quadro 3 - Descrição das métricas implementadas no SNSAnalyser

Além de oferecer essas métricas pré-definidas, a solução proposta inclui duas outras funcionalidades para permitir que análises mais avançadas sejam feitas. Primeiro, a solução inclui a possibilidade de incorporar no sistema novas métricas que devem ser implementadas seguindo uma interface definida. Assim, a solução fica passível de evolução e aprimoramento da análise através de qualquer medida que seja implementada e “plugada” no sistema, permitindo que a rede seja analisada em relação a qualquer informação de interesse do pesquisador.

A outra funcionalidade requerida é a inclusão de pelo menos um formato de exportação que seja reconhecido pelas ferramentas UCINet e NetMiner, possibilitando que uma análise mais abrangente seja feita através de um desses sistemas. Isso é uma característica comum nas ferramentas analisadas e também existe na ferramenta construída.

Assim, a ferramenta construída ao longo deste trabalho permite a análise através de suas métricas pré-definidas, métricas customizadas desenvolvidas pelos usuários ou exportação para análise em outras ferramentas.

2.6 CONSIDERAÇÕES FINAIS

Neste capítulo foram apresentadas a fundamentação teórica e a revisão de literatura sobre o tema Análise de Redes Sociais. Importantes conceitos relacionados a tal área de estudo foram apresentados.

As escolas da análise de redes sociais, aplicações, benefícios e a seleção da amostra foram abordados, bem como a análise a partir de métricas. Algumas ferramentas são avaliadas e um comparativo entre elas é realizado na seção 2.4.2. Ao final, são apresentados os requisitos para a solução de análise de redes sociais proposta neste trabalho.

A fim de subsidiar o entendimento técnico sobre a extração das redes sociais que serão análisadas conforme foi visto neste capítulo, o próximo capítulo aborda esse assunto e apresenta uma proposta de solução de extração à qual se integra a proposta de análise aqui exposta.

3 EXTRAÇÃO DE REDES SOCIAIS DA INTERNET

A extração automática de informações da Internet é um assunto relativamente recente. Hope e outros (2006) definem três possíveis abordagens para extrair e combinar redes sociais e propõem um método que combina todas elas.

A primeira abordagem é baseada em técnicas de mineração, tornando possível a detecção automática de relacionamentos a partir de várias fontes de informação, como arquivos de e-mail, dados de agendamento e informações de citação na Internet (GRUDIN, 1994).

Na área de redes sociais, vários estudos têm abordado a extração de redes sociais automaticamente a partir de diversas fontes de informação. Mika (2005) desenvolveu um sistema de extração, agregação e visualização de redes sociais on-line chamado Flink para uma comunidade de web semântica na qual as redes sociais são obtidas utilizando páginas da *web*, mensagens de correio eletrônico e publicações. Utilizando uma abordagem semelhante, Matsuo e outros (2006) desenvolveram um sistema chamado Polyphonet. Em consonância com esses estudos, vários estudos têm explorado extração automática de redes sociais a partir da Internet (ADAMIC e outros, 2003; CULOTTA e outros, 2004; HARADA e outros, 2004; KAUTZ e outros, 1997).

O segundo meio utilizado para extrair redes sociais é baseado na interação entre os utilizadores do mundo real (por exemplo, comunicação face-a-face) em comunidades (HOPE e outros, 2006, p. 2). A extração é feita com a captura das interações entre os usuários utilizando equipamentos que detectam e registram automaticamente determinadas ações das pessoas monitoradas (PENTLAND, 2005).

Outra abordagem se baseia na interação entre os usuários e um sistema, onde os usuários podem descrever sua própria rede social. Essa descrição das informações das pessoas e suas relações com as outras é denominado por Hamasaki e outros (2006) de *Friend-of-a-Friend* (FOAF). A possibilidade de coletar informações FOAF e obter uma rede FOAF é abordada também em Finin e outros (2005).

Todas as três abordagens têm seus prós e seus contras. O quadro abaixo apresenta as principais vantagens e desvantagens de cada uma.

	Vantagens	Desvantagens
Mineração na web	- Fornecem uma visão boa das pessoas proeminentes. - Relacionamentos baseados em registros de informações reais fornecem dados confiáveis	- Não registram apropriadamente relacionamentos de novatos, estudantes e algumas outras pessoas "normais". - Dificuldade na definição de escopo da rede social.
Monitoramento de ações dos usuários (através de equipamentos)	- Fornecem uma visão relativamente boa das pessoas importantes. - Relacionamentos baseados em registros de informações reais monitoradas fornecem uma confiabilidade maior que os dados cadastrados por livre e espontânea vontade e menor que os dados reais não monitorados.	- Vinculado a características específicas dos dispositivos: pode haver erros da detecção, limitação de espaços da detecção e uso tendencioso dos usuários. - Possibilidade de limitação/resistência na obtenção de usuários em função da necessidade de monitoramento.
Interação com um sistema (SNSs)	- Simplicidade na obtenção dos relacionamentos, uma vez que é o próprio usuário que entra com essa informação no sistema.	- Forte dependência do usuário no que tange a quais relacionamentos devem ser registrados pode provocar discrepâncias entre os dados registrados por eles.

Quadro 4 - Vantagens e desvantagens de cada método de extração de rede social

A opção proposta neste trabalho é uma mistura da primeira abordagem com a terceira. Assim, a idéia aqui é extrair redes sociais através da mineração das informações já armazenadas em serviços de rede social acessível através da internet. A fim de se extrair redes sociais de grandes sites de relacionamento, a seção seguinte discute algumas das principais estratégias para isso.

3.1 ESTRATÉGIAS PARA EXTRAIR REDES SOCIAIS DE GRANDES SITES DE RELACIONAMENTO

Entende-se por site de relacionamento um sistema de rede social disponível na *web*. A extração de uma rede social a partir de um site de relacionamento consiste na extração dos nós e arestas que formarão a rede social, onde os nós são usuários do site de relacionamento e as arestas, ligações entre os usuários registradas no site. A extração das arestas é referenciada nesse documento como extração de relacionamentos. Existem duas maneiras de extrair redes sociais a partir de um site de relacionamento. A primeira delas é a utilização de API e não pode ser utilizada em qualquer sistema de rede social, pois depende da empresa que mantém o site relacionamento na *web*.

A utilização de uma API fornecida pelo próprio sistema de rede social é em muitos casos, a forma preferida de se extrair os relacionamentos de um usuário de site de relacionamento. Isso porque, sem dúvida, essa tende a ser a abordagem de melhor desempenho na realização desta extração. Entretanto o uso de uma API tem muitas outras limitações que às vezes fazem com que sua adoção não seja tão interessante.

O principal problema é que cada rede social possui a sua API que pode nada ter a ver com a API de outra rede social, desenvolvida por outra empresa. Isso é muito ruim quando se quer desenvolver ferramentas genéricas que funcionem independente da rede social. Além disso, pode ser difícil para o desenvolvedor aprender uma API para cada rede social que ele queira utilizar.

No início de novembro de 2007, a Google™ lançou a API que será utilizada no Orkut™, seu site de rede social. A OpenSocial é uma API que está sendo desenvolvida pela Google™ em conjunto com membros da comunidade. O objetivo é que qualquer site de rede social possa implementar a API e hospedar aplicações de redes sociais desenvolvidas por terceiros. Segundo a Google™, existem muitos sites implementando o OpenSocial, incluindo Engage.com, Friendster, hi5, Hyves, imeem, LinkedIn, MySpace, Ning, Oracle, Orkut, Plaxo, Salesforce.com, Six Apart, Tianji, Viadeo e XING.

Esta pode ser uma boa iniciativa, mas como seu lançamento ainda está muito recente, a OpenSocial ainda apresenta muitos problemas técnicos e não é funcional ou não está disponível na maioria dos sistemas de rede social disponíveis na *web* atualmente.

A segunda questão com relação ao uso de APIs é que as funcionalidades permitidas são aquelas que a empresa que forneceu a API disponibilizou em sua interface. Isso faz com que muitas ações disponíveis para o usuário no site não possam ser realizadas pela aplicação que utiliza a API, pois os procedimentos necessários para ela não foram disponibilizados.

A outra forma de extrair redes sociais da *web* é a utilização de *Web Spiders. Spider* é apenas uma das denominações utilizadas para aplicações que funcionam como um robô acessando páginas disponíveis na *web*. Além desse termo, também são utilizados *Web Crawlers* e, em menor quantidade, *Web Robots* ou *Worms*.

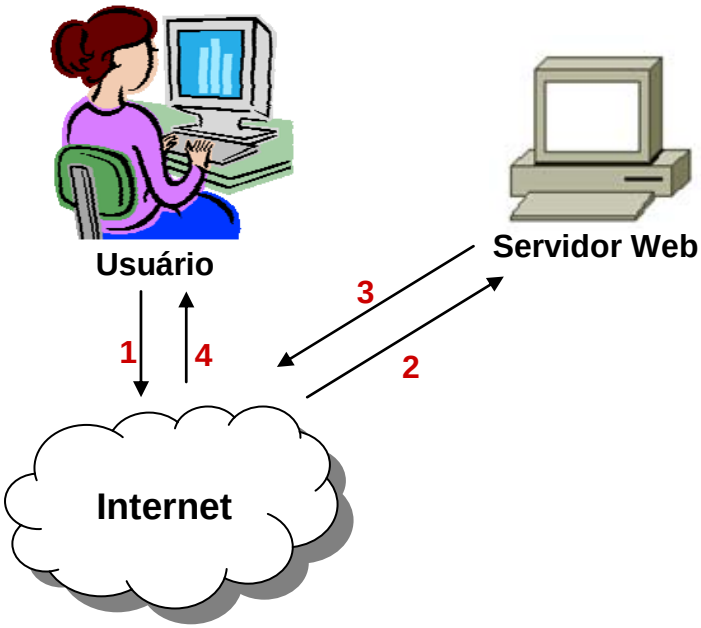


Figura 8 - Trajeto de uma requisição de um usuário até um servidor web

Uma requisição comum que parte do *browser* de um usuário e segue até um servidor *web* acontece conforme é ilustrado na Figura 8. Ou seja, a requisição do usuário percorre a internet até chegar ao servidor *web* em questão, que realiza o processamento necessário e retorna o hipertexto solicitado para o usuário. Já com a utilização de um *web crawler*, esse caminho torna-se um pouco diferente, como mostra a Figura 9.

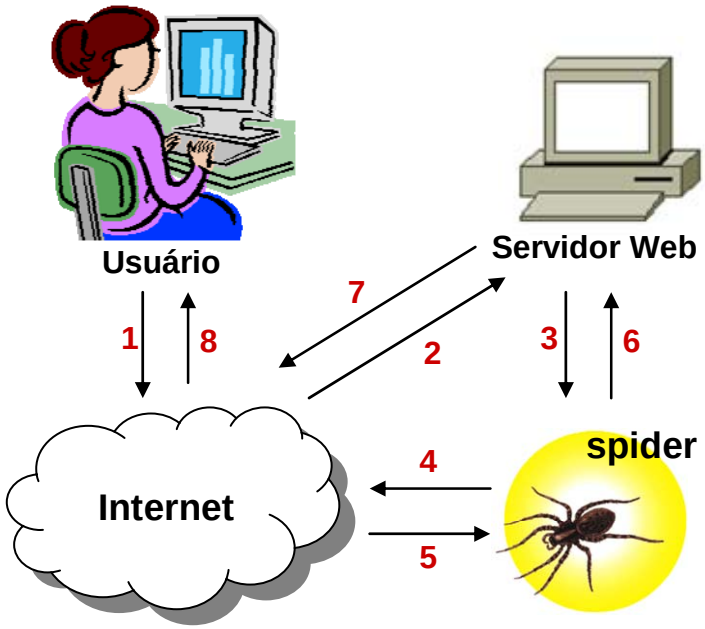


Figura 9 - Trajeto de uma requisição de um usuário até um servidor web que utiliza spider

Nesse modelo, quando a requisição chega ao servidor *web*, ele ativa o *spider* que acessa a internet para obter as informações desejadas. Uma vez feito isso, o *spider* devolve o conteúdo ao servidor *web*, que, só então, gera o hipertexto que será devolvido ao usuário.

A primeira geração de *crawlers* (HEINONEN; HATONEN; KLEMETTINEN, 1996), em cuja maioria dos algoritmos atuais de busca se inspira, baseia-se principalmente em algoritmos tradicionais de estudos de grafos, como busca em profundidade e busca em largura, para indexar a *web*. O principal problema do *spider* é o seu desempenho inferior, quando comparado com o acesso direto utilizando a API do fornecedor. Como ele realmente simula o acesso feito pelo usuário a páginas disponíveis na *web*, as informações são obtidas a partir da camada mais alta da aplicação: a camada de apresentação. Desta forma, o *spider* apresenta um processamento mais custoso tanto para as máquinas que estão extraindo a informação, pois elas fazem todo o processamento até a interface com o usuário, o que demanda tempo, quanto para o sistema de rede social que está sendo alvo do *spider*, pois ele receberá um número de requisições por segundo muito mais alto que um ser humano seria capaz de enviar.

Entretanto, pelo fato de trabalhar sobre o protocolo HTTP, ao invés de APIs específicas, a utilização do *spider* possibilita que desenvolvedores criem uma infraestrutura na qual é permitido o desenvolvimento de ferramentas de extração mais genéricas e independentes das redes sociais como linguagens de programação específicas para navegação e extração de dados a partir de páginas HTML. Este trabalho utiliza essa tecnologia e o capítulo 4 contém uma seção dedicada à abordagem desse assunto.

3.1.1 Definição de escopo

Uma das formas de se classificar uma rede social é com relação ao seu escopo. Segundo esse critério, as redes podem ser: totais, também conhecidas como sociocêntricas, que são aquelas que possuem um conjunto completo de relacionamentos em uma unidade de análise (empresa, família, departamento, etc.); egocêntricas, em que a maioria dos nós está conectada a nós simples ou individuais; e redes de sistemas abertos, que são redes em que as fronteiras não são necessariamente claras (HANNEMANN, 2001).

O tamanho do conteúdo público indexável da *World Wide Web* ultrapassou um bilhão de documentos (IICM, 2008) e, até o momento, o crescimento não mostra nenhum sinal de redução de ritmo. Mesmo usando sistemas no estado da arte, como o AltaVista *Scooter*, que

se reporta realizar o rastreamento de dez milhões de páginas por dia, um exaustivo rastreamento da *web* pode demorar semanas (DILIGENTI e outros, 2000).

Os grandes sistemas de redes sociais disponíveis na *web* possuem bases de dados com muitos milhões de usuários. Segundo dados encontrados em WIKIPEDIA (2007), o Orkut™ possuía em 20 de agosto de 2007 mais de sessenta milhões de usuários cadastrados (precisamente, 68.182.265 usuários). O Myspace™ (MYSPACE, 2008), quinto site mais acessado do mundo e o mais popular dentre os de rede de relacionamento em março de 2008 segundo o Alexa (2008), tinha cerca de 110 milhões de membros cadastrados em novembro de 2007, segundo dados publicados no portal de notícias da Globo™ (G1, 2008).

Diante desse contexto, muitas vezes torna-se uma operação muito complicada extrair todos os relacionamentos existentes em uma rede social como essas. Supondo que cada membro tenha 100 relacionamentos, o que verdadeiramente não é um número muito grande, implicaria que numa rede como a do Orkut™ seria necessária a obtenção de quase sete bilhões de relacionamentos, o que poderia levar semanas, meses, ou até anos para se concretizar, dependendo do parque tecnológico disponível para essa operação. Além disso, com o dinamismo característico do crescimento desses sistemas, as chances de, ao final da extração, os dados estarem desatualizados são grandes.

Na medida em que as máquinas de busca começaram a ter dificuldades em percorrer toda a Internet com seu conteúdo dinâmico e crescente, alguns trabalhos foram realizados no intuito de prover soluções para realizar as mesmas tarefas com foco mais direcionado. É o caso, por exemplo, do *Context Focused Crawler* de Diligenti e outros (2000), que propõem que o rastreamento dos robôs seja sensível ao contexto das buscas. Nesse cenário, ao propor a extração de redes sociais a partir de grandes sites de relacionamento, não se pode deixar de pensar em estratégias de redução de escopo.

Algumas formas de redução de escopo podem ser utilizadas. A primeira delas é definir um tempo ou quantidade de relacionamentos para uma amostragem da rede. Esta técnica é muito boa quando se tem um ponto de partida interessante na rede ou quando se tem um tempo de extração ou número de relacionamentos bem definidos antes da coleta. Para esta técnica, parte-se de um ou mais pontos de partida e segue-se obtendo os relacionamentos de cada membro num efeito cascata (primeiro os amigos do ponto de partida, depois os amigos desses amigos e assim sucessivamente). No entanto, na maioria das vezes, esse tipo de

extração resultará numa rede que não necessariamente apresenta as características da rede completa e, por isso, o uso dessa técnica deve ser bem avaliado para que os dados analisados não deformem a visão correta da realidade.

Outras duas técnicas que enfocam nichos de usuários mais específicos podem ser utilizadas e apresentam redes resultantes muitas vezes mais interessantes do que uma simples amostra da rede completa. A primeira delas depende da existência de uma base prévia de usuários com a informação que associa esses usuários a um membro do sistema alvo de onde serão extraídos os relacionamentos. Por exemplo, pode-se montar uma base de dados onde o usuário 1 de um sistema local, corresponde ao usuário 1000 no Serviço de Rede Social alvo (do inglês, *Social Network Service* – SNS), o usuário 2 corresponde ao 30 e assim por diante. A partir daí, pode-se realizar a extração dos relacionamentos registrados no site de relacionamento entre os usuários da base pré-existente. Ou seja, pode-se formar, a partir de uma base de usuários, a rede de relacionamentos que eles formam, extraindo os relacionamentos existentes entre eles de um site de relacionamento.

Por fim, a outra forma de se obter uma rede de escopo menor, mas também com um bom foco é a extração a partir de comunidades ou grupos existentes nos sites de relacionamento. Comunidades são grupos formados dentro dos sites de relacionamento compostos por usuários que têm características, perfis ou interesses em comum. A extração de comunidades consiste em extrair primeiramente os membros da comunidade, ou seja, os atores ou nós da rede. Em seguida, extraem-se os relacionamentos entre os membros, completando assim a extração da rede social. Optando por extrair os membros de mais de uma comunidade antes de extrair os relacionamentos, pode-se criar, inclusive, uma rede social a partir de mais de uma comunidade do site de relacionamento. Isso é bastante interessante quando as comunidades são semelhantes. Dessa forma, consegue-se extrair comunidades reais a partir de comunidades virtuais. Exemplificando: No Orkut™ existe uma comunidade chamada “Eu amo açaí” e outra “Eu adoro açai”. É interessante extrair os membros dessas duas comunidades (provavelmente pessoas que adoram açaí) e, a partir daí, obter os relacionamentos entre esses membros para criar a rede social resultante e poder analisar todos esses usuários em conjunto, como se fossem membros de uma única comunidade. Atuando assim, essa técnica se revela bastante adequada para se estudar diversas pequenas redes dentro de redes maiores com o foco em algum assunto de interesse do pesquisador.

3.1.2 Tipos de busca

Para realizar a extração, independente de ser através da utilização de uma API ou por meio de um *spider*, dois tipos de busca de dados podem ser utilizados. O primeiro deles e mais simples é a busca serial ou *simple thread*. A obtenção de dados de forma serial significa que um relacionamento será obtido de cada vez, um após o outro. Essa é uma estratégia simples e que funciona perfeitamente para uma quantidade não muito grande de dados.

Na medida em que o volume de dados a serem extraídos cresce, torna-se necessário utilizar uma estratégia de busca de dados paralela, aqui denominada também como *multi thread*. Nessa estratégia, mais de um processamento estará sendo feito ao mesmo tempo, de forma que, enquanto a *thread* 1 está obtendo os relacionamentos do usuário 1, a *thread* 2 estará rodando e extraindo os relacionamentos de um outro usuário.

Essa técnica garante uma velocidade maior na extração de dados e por isso é indicada em situações de grandes volumes de dados. Entretanto, a implementação da busca precisa de um mecanismo de armazenamento dos usuários que garanta o controle de concorrência.

Enquanto, com a extração serial, a *thread* única precisa apenas obter o próximo usuário em uma fila para extrair os relacionamentos dele, na busca com múltiplas *threads*, é necessária uma *flag* que registre o status de cada usuário, informando se ele está aguardando na fila, se está sendo utilizado ou se já teve seus relacionamentos extraídos. Além disso, vale ressaltar a importância do uso de um mecanismo de tranca ou semáforo no momento da leitura e escrita dessa informação.

3.1.3 Cuidados no armazenamento

Durante a extração dos dados do site de relacionamento, é preciso ir armazenando os relacionamentos em uma base de dados local. Nesse momento, alguns cuidados precisão ser tomados para se evitar redundâncias e possibilitar que suas consultas apresentem um desempenho aceitável mesmo quando o volume de dados for muito grande.

A primeira preocupação é com a eliminação de duplicidade entre os registros. Ao se obter os relacionamentos do indivíduo A numa rede social, como um dos resultados da extração virá que A se relaciona com B. Da mesma forma, ao extrair os relacionamentos de B, será obtido que B se relaciona com A. Nas redes sociais em que essas duas informações

representam a mesma coisa, se não for tomado o cuidado de registrar apenas um dos dois relacionamentos, o banco local irá ficar com o dobro de registros e nos casos em que o volume de dados é muito grande isso faz muita diferença.

Uma forma simples de garantir isso é proposta pelo autor deste trabalho e denominada de “Menor Primeiro”. Essa técnica consiste em fazer uma comparação entre os identificadores dos usuários antes de inserir o relacionamento no banco de dados. Após a comparação, ordena-se o relacionamento de forma que o menor identificador fique na primeira posição e o maior, na segunda. Como o nome já diz, a primeira posição do par do relacionamento irá pertencer sempre ao registro de menor identificador. Por exemplo, se num relacionamento entre os usuários A e B o identificador de A for menor que o de B, o relacionamento armazenado será AB, caso contrário, será BA. Pode-se questionar se, de qualquer forma, não será necessário verificar a existência do relacionamento antes de inseri-lo. É verdade, mas se esse controle não fosse realizado seriam necessárias duas consultas para garantir a unicidade, pois seria preciso verificar se AB existe e se BA existe. Neste caso, apenas a verificação com o menor primeiro é necessária. E mais, caso o dispositivo de armazenamento possua mecanismos que garantam unicidades de registro, como a chave primária de um banco de dados, ele mesmo fará a verificação e lançará um erro que pode ser devidamente tratado ou ignorado caso se tente inserir um registro repetido.

Outro ponto importante para se preocupar com relação ao armazenamento das informações refere-se à distribuição, o que poderá garantir a escalabilidade da sua base de informações. A depender do dispositivo de armazenamento utilizado, poderão existir limitações de tamanho ou quedas de desempenho quando o volume de dados ficar muito grande. Muitos gerenciadores de bancos de dados registram em suas especificações que não existe limite de registros por tabela, como é o caso do SQL Server 2005 (MSDN, 2008) e do PostgreSQL (POSTGRESQL, 2008). Entretanto, na prática, sabe-se que trabalhar com tabelas muito grandes, com mais de 20 milhões de registros, por exemplo, torna os índices menos eficientes e as consultas ficam bastante lentas. Neste caso, deve-se pensar em distribuição.

Apesar de alguns gerenciadores de bancos de dados já oferecerem esse recurso, existem formas não muito complicadas de se distribuir o armazenamento por meio de verificações simples. Um método proposto pelo autor desse trabalho e denominado “Último do Primeiro” é a criação de um *hash* que, a partir do relacionamento, possa se chegar à tabela em que ele está armazenado. O método, que pode ser utilizado nos casos em que os

identificadores dos usuários possuam apenas algarismos, consiste em se basear no último algarismo do primeiro identificador para determinar em qual das dez partições se encontra o relacionamento em questão. Desta forma, obtém-se um número entre zero e nove e o mecanismo de distribuição pode ter até dez tabelas, onde cada uma delas é mapeada para um desses dígitos. Para identificadores alfanuméricos, pode-se realizar uma adaptação ao método proposto e utilizar um conjunto de caracteres para cada tabela. Para ampliar a distribuição, outra alteração na idéia inicial pode ser feita como, por exemplo, a utilização dos dois últimos caracteres. Assim, pode-se distribuir em até 100 tabelas distintas.

O principal problema desse tipo de distribuição é quando se deseja obter todos os relacionamentos de um determinado usuário. Se isso for necessário, terá que ser feita uma busca em todas as tabelas, uma vez que seus amigos estarão distribuídos entre elas, o que tornará a consulta bem mais custosa. Para evitar isso, uma proposta deste trabalho é que, nos casos em que exista esse requisito, se realize a replicação dos dados. Ou seja, todos os relacionamentos de A estariam na tabela em que a está registrado e todos os relacionamentos de B estariam na tabela em que B está registrado. Assim, o banco armazena tanto o relacionamento AB como BA. No momento da inclusão pode-se continuar usando o método do “Último primeiro”, mas neste caso, deve-se inserir nas duas tabelas resultantes do *hash*. Desse modo, a quantidade de registros duplica, mas, em compensação, o desempenho na obtenção dos relacionamentos de um usuário melhora bastante. Em função disso, deve-se analisar com cautela para escolher o modelo que melhor se adequa a cada caso. Como a distribuição contribui para diminuir a quantidade de registros em uma única tabela, replicar os dados, pode se tornar uma opção aceitável para os casos em que o volume de dados é grande, a distribuição é necessária e sem isso o desempenho é muito pequeno.

3.1.4 Evitando bloqueios ao sistema

A coleta de informações muitas vezes é combatida pelos sites da Internet que estão sendo alvos da extração. Muitos deles monitoram as requisições que estão sendo feitas aos seus servidores tanto em relação aos usuários de suas redes que estão requisitando as páginas, quanto ao IP da máquina de onde partiu a requisição. Requisições feitas por um mesmo usuário, partindo de um mesmo IP e com intervalos entre elas muito pequenas são suspeitas e muitas vezes impossíveis de serem realizadas por um ser humano.

Sabendo disso, medidas precisam ser tomadas para evitar bloqueios ao sistema que pretende extrair os dados de SNSs. Não se tem notícia de bloqueios a sistemas que utilizam as APIs disponibilizadas pelos sites de rede social, então as estratégias descritas a seguir são principalmente aplicadas quando se está utilizando um *spider*.

Para evitar sobrecarga nos sites de relacionamento, deve-se dimensionar o spider para executar com velocidade próxima a de um ser humano. Neste sentido, faz-se necessária a inclusão de um *delay* entre as requisições. Enquanto a maioria das aplicações trabalha para aumentar seu desempenho, com a utilização dos *spiders* busca-se exatamente o contrário. Caso não seja colocado um intervalo de "descanso" entre uma requisição e outra, mecanismos de monitoramento dos servidores do site podem perceber que um ser humano não consegue acessar suas páginas de forma tão rápida e considerar que aquelas requisições vêm de um robô. Essa estratégia trata-se de uma degradação intencional do desempenho no intuito de aumentar a semelhança da navegação do *spider* com uma navegação humana. O impacto dessa estratégia no desempenho é diretamente proporcional à quantidade de requisições necessárias. Assim, geralmente, quanto maior o volume de informações que precisam ser extraídas, maior o número de requisições necessárias e, consequentemente, maior o impacto no desempenho.

Outro recurso que pode ser utilizado em conjunto com o *spider* na extração dos relacionamentos é a utilização de mais de um usuário navegador. O usuário navegador é o usuário utilizado para fazer *log in* na rede social da qual se pretende extrair os relacionamentos. Principalmente, quando é utilizado um esquema *multi thread* de busca de dados, a utilização de apenas um usuário da rede social pode ser uma estratégia arriscada. Isso porque as redes também monitoram a quantidade de requisições por usuário e descobrem essas contas que acessam muito mais o SNS do que humanos comumente acessam. A utilização de mais de um usuário navegador serve para diluir o acesso entre esses usuários.

Por fim, além dessas estratégias, pode-se partir para a utilização de *proxies*, uma vez que os IPs também são alvo das análises dos sites alvo. Ou seja, apesar da utilização da Internet através de *lan houses*, escolas, faculdades e empresas implicar no acesso de muitas pessoas a diversas páginas na *web* pelo mesmo IP, ainda assim, os sites de relacionamento conseguem utilizar o IP para verificar a existência de robôs acessando suas páginas. Para resolver, esse problema, pode-se adotar outras máquinas como *proxies* e fazer com que a extração se dê através dos diferentes IPs de cada *proxy*, evitando assim mais uma chance de

bloqueio. No capítulo 4, será abordada a utilização do Power Proxy e do Power Manager para realizar essa função.

3.2 PROPOSTA DE SOLUÇÃO PARA A EXTRAÇÃO DAS REDES SOCIAIS

Após análise das estratégias que podem ser utilizadas na extração de redes sociais a partir de grandes sites de relacionamento, a primeira decisão fica entre a utilização ou não de *web crawlers* como ferramenta de navegação e obtenção das informações da *web* em prejuízo do uso de APIs disponibilizadas pelas redes originais.

Para essa decisão, alguns fatores foram cruciais na avaliação. O primeiro deles é o fato de nem todos os grandes sites de relacionamento disponibilizarem APIs para desenvolvimento de aplicações, incluindo o mais acessado do Brasil em abril de 2008, segundo o Alexa (2008), o Orkut™. O mais acessado nesse período segundo o Alexa (2008), o MySpace™, e outros grandes como o hi5 também não oferecem essa funcionalidade, apesar de alguns deles informarem que em pouco tempo isso será possível.

O outro fator que não favoreceu o uso de APIs é o fato de que cada rede social pode oferecer uma API para o desenvolvimento de aplicações. Apesar da iniciativa da Google™ com o OpenSocial, algumas das grandes redes já adotam uma API proprietária muito difundida e não devem abrir mão dela, como é o caso do FaceBook™ (FACEBOOK, 2008).

Por essa ótica, a adoção de um *web crawler* para extrair redes sociais de sites de relacionamento torna-se mais adequada. Ao adotar essa opção, a construção de uma ferramenta genérica que navegue em qualquer site torna-se mais simples. Além disso, existem linguagens de programação especificamente projetadas para facilitarem ainda mais essa questão, permitindo que o usuário da ferramenta acrescente quantas novas redes interessarem, bastando, para isso, incluir os scripts necessários.

Ainda analisando esse ponto, o único contra encontrado na utilização do *crawler* seria o seu desempenho inferior, quando comparado com o acesso direto utilizando a API do fornecedor. Este quesito, entretanto, é considerado menos relevante neste caso, uma vez que a ferramenta terá um escopo de extração muito bem definido, o que implica em uma extração menos custosa.

Como hoje já existem muitos sites de relacionamento que, através das interações de seus usuários já conseguiram estabelecer relacionamentos entre mais de 40 milhões de usuários no mundo, o SNSAnalyser não irá se propor a extrair uma rede social a partir de diversas fontes de informações, mesmo isso sendo possível. A proposta da ferramenta é que sua utilização seja principalmente na extração de redes sociais que serão criadas a partir dos relacionamentos registrados em sites de relacionamento.

Considerando essa proposta, o escopo de navegação já se reduz dos bilhões de documentos existentes na Internet (IICM, 2008), para os milhões de usuários das redes sociais. Entretanto, esse escopo ainda está muito grande.

A obtenção de uma parte da rede realizando a extração por um período de tempo ou para uma quantidade de usuários foi descartada em função da rede resultante não apresentar nenhuma característica de similaridade forte que justificasse a escolha desses usuários.

O filtro por uma base pré-existente também não se mostrou uma boa opção pelo fato de que nem sempre a pessoa ou empresa interessada na análise de uma sub-rede de um site de relacionamento irá dispor de uma base prévia de usuários.

Entre as estratégias que poderiam ser utilizadas para restringir o escopo, a extração de redes sociais a partir de comunidades é a mais atrativa, pois possibilita uma análise em nichos de mercado e grupos de usuários com características ou interesses comuns.

Um ponto importante com relação ao filtro de comunidades é que o SNSAnalyser permite sua utilização quantas vezes forem necessárias antes da obtenção dos relacionamentos. Dessa forma, pode-se obter, por exemplo, os membros da comunidade “Eu amo açaí”, depois os da “Eu adoro açaí” e somente após isso, partir para a obtenção dos relacionamentos e da rede social resultante – a rede das pessoas que adoram ou amam açaí.

Seguindo nas estratégias, a solução escolhida adota ambos os tipos de busca, o *simple thread* e o *multi thread*, mas em funcionalidades distintas. Em função da opção de extração por uma metodologia na qual primeiro são obtidos os membros da rede para depois buscar os relacionamentos, o SNSAnalyser realiza uma busca serial na extração dos membros, uma vez que na maioria dos casos será necessária a execução de apenas um Power Script para realizar essa tarefa. Já para a extração de relacionamentos, a solução proposta foi a possibilidade de

configuração da ferramenta para indicar a quantidade de *threads* que serão instanciadas nesse procedimento.

Com relação ao armazenamento, alguns cuidados foram tomados no intuito de evitar duplicidade de registros, mas também visando manter o desempenho da aplicação alto. A primeira estratégia foi a utilização do critério do Menor Primeiro ao armazenar os relacionamentos extraídos, o que garantiu a unicidade dos registros. Já no registro de distâncias entre nós, optou-se por registrar tanto as distâncias de A até B, como a de B até A para melhorar a performance dos algoritmos. Já no quesito escalabilidade e distribuição, o SNSAnalyser não se propôs a implementar uma solução em código, uma vez que o objetivo do sistema é realizar extrações de redes sociais com foco bem definido. Além disso, a adoção do SQL Server 2005 como banco de dados, já possibilita o particionamento de tabelas e índices, o que facilmente resolveria o problema se surgisse essa necessidade (MSDN, 2008).

Por fim, com a adoção do *spider* como ferramenta de extração de informações da *web*, algumas estratégias anti-bloqueio precisaram ser utilizadas. A primeira delas, a utilização de *delay*, é usada a qualquer momento no SNS Analyser, a critério do desenvolvedor do Power Script, pois o próprio Power Script dispõe de comandos para tal funcionalidade. Além disso, a fim de evitar bloqueios ao sistema, o SNSAnalyser permite que uma rede social possa estar associada a mais de um usuário navegador, ficando a cargo do usuário o cadastro de quantos usuários navegadores achar necessário. Vale ressaltar que é importante que a criação desses usuários tenha sido feita de acordo com os termos de uso do site de relacionamento. A utilização de outras máquinas como *proxy* também é permitida no sistema, através da configuração e utilização do Power Proxy e do Power Manager. Ambos serão mais bem detalhados no capítulo 4.

3.3 CONSIDERAÇÕES FINAIS

Neste capítulo foram abordados os aspectos referentes à recuperação de informações da Internet, no que tange à extração de redes sociais. Um resumo de pesquisas e trabalhos encontrados durante a revisão da literatura relacionada a tal tema foi apresentado, bem como uma análise de vantagens e desvantagens de cada método passível de utilização e estratégias que podem ser aplicadas em todas as fases desse processo.

O capítulo culmina com a apresentação de uma proposta de solução para a extração de redes sociais a partir de sites de relacionamentos, onde são abordados os motivos pelos quais cada estratégia foi adotada.

O próximo capítulo descreve o SNSAnalyser, ferramenta desenvolvida com base na proposta de solução para extração e análise de redes sociais a partir de comunidades existentes em sites de relacionamento. Tal sistema informatizado possui seus alicerces na extração de informações da *web* e na análise de redes sociais, contemplando funcionalidades que implementam as propostas deste trabalho nestas áreas.

4 FERRAMENTA SNSANALYSER

Após o estudo descrito nos capítulos 2 e 3 deste documento, tornou-se possível propor uma solução para a extração e análise de redes sociais a partir de comunidades existentes em sites de relacionamento.

O presente capítulo apresenta o SNSAnalyser, uma ferramenta construída à luz da solução proposta pelo autor deste trabalho. O nome SNSAnalyser surgiu da união dos termos SNS, que significa Social Network System (sistema de redes sociais) e Analyser que é traduzida do inglês para o português como analisador.

Com uma abordagem diferente das ferramentas pesquisadas, o SNSAnalyser foi desenvolvido no intuito de apoiar as empresas e os profissionais que se interessam em avaliar a rede social composta por usuários de grandes sites de relacionamento com interesses ou características similares, para selecionar os que, dentro de um determinado perfil, se destacam dentro da rede. A identificação de indivíduos importantes em comunidades bem definidas tem muitas aplicações práticas em áreas como análise de mercado, gestão de conhecimento, marketing, seleção de RH, investigação criminal, antropologia, entre muitas outras.

Com o objetivo de proporcionar um melhor entendimento sobre a ferramenta em questão, este capítulo dedica-se ao detalhamento do SNSAnalyser, iniciando com a abordagem dos seus principais requisitos funcionais, seguindo com a infra-estrutura utilizada, sua especificação técnica e, ao final, uma explanação minuciosa sobre suas funcionalidades.

4.1 REQUISITOS FUNCIONAIS E NÃO-FUNCIONAIS

Como parte integrante da solução discorrida no presente trabalho, o SNSAnalyser atende aos seguintes requisitos funcionais relativos à proposta:

1. Possibilitar a extração de redes sociais a partir de qualquer site de relacionamento, utilizando filtro de comunidade.

2. Permitir a análise matemática das redes sociais extraídas, através da utilização das métricas pré-definidas selecionadas na seção 2.5 deste documento, métricas

customizadas inseridas na ferramenta pelo usuário e exportação dos dados extraídos para as ferramentas UCINet, NetMiner e Excel™.

Adicionados aos requisitos funcionais acima, alguns requisitos não-funcionais também tiveram sua solicitação atendida, são eles:

1. Possibilitar a realização de buscas *multi thread*.
2. Evitar duplicidade de registros, exceto quando utilizada em função da melhoria do desempenho dos algoritmos de cálculo de métricas.
3. Acessar as páginas dos sites de relacionamento com velocidade próxima a do ser humano, de modo a evitar bloqueios a robôs por parte dos sites de relacionamento.
4. Estar disponível através da *web*.

Referenciando as estratégias adotadas, o atendimento do primeiro requisito funcional implica a opção pela utilização do *spider* e também diminui o escopo de extração para as comunidades existentes dentro das redes sociais. Cada um dos requisitos não-funcionais também é atendido pela estratégia proposta na seção 3.2 deste documento.

Os requisitos, então, foram subdivididos em casos de uso, a saber:

Possibilitar a extração de redes sociais a partir de qualquer site de relacionamento, utilizando filtro de comunidade:

1. Manter rede social;
2. Extrair membros da rede social;
3. Extrair relacionamentos dos membros obtidos;
4. Manter dados extraídos.

Permitir a análise das redes sociais extraídas, através da utilização de métricas pré-definidas e customizadas:

1. Calcular métricas pré-definidas;
2. Exibir resultados de cálculos de métricas pré-definidas;

3. Manter métricas customizadas;
4. Calcular métricas customizadas;
5. Exibir resultados de cálculos de métricas customizadas;
6. Exportar dados para o UCINET;
7. Exportar dados para Excel ™;

Utilizar algum um tipo de proteção anti-bloqueio:

1. Manter usuário navegador;

Os requisitos não funcionais não podem, em sua maioria, ser mapeados para casos de uso específicos. Seu atendimento se dá de forma transversal aos requisitos funcionais.

Possibilitar a realização de buscas *multi thread*:

1. Durante a implementação do caso de uso “Extrair relacionamentos dos usuários obtidos” são incluídos procedimentos que permitem extrair os relacionamentos de usuários diferentes de forma separada, independente e paralela.

Evitar duplicidade de registros, exceto quando utilizada em função da melhoria do desempenho dos algoritmos de cálculo de métricas:

1. No desenvolvimento da funcionalidade de extração de relacionamentos, são utilizadas as estratégias definidas no capítulo 3 para evitar duplicidade de registros.
2. No cálculo de métricas pré-definidas, o armazenamento dos registros segue a orientação para que apresente maior desempenho nas consultas.

Navegar com comportamento próximo ao do ser humano, de modo a evitar bloqueios a robôs por parte dos sites de relacionamento:

1. Para se aproximar ao comportamento humano, atrasos são incluídos entre as requisições realizadas aos sites de relacionamento, o que é possível através da infra-estrutura utilizada.

2. Para atendimento desse requisito, também é desenvolvido o caso de uso “Manter usuário navegador” que possibilita o cadastro de quantos usuários forem necessários para a navegação nos sites de relacionamento.

Estar disponível através da *web*:

1. Este requisito é atendido através da escolha da linguagem de programação utilizada.

Visando o atendimento aos requisitos acima abordados, as funcionalidades da ferramenta SNSAnalyser foram especificadas e suas características técnicas, delineadas, conforme abordado na seção subseqüente.

4.2 CARACTERÍSTICAS TÉCNICAS

A ferramenta SNSAnalyser foi desenvolvida seguindo os requisitos funcionais discorridos anteriormente. Foi analisada qual a melhor forma de atender cada um dos requisitos, o que gerou o conjunto de características técnicas da referida ferramenta.

No processo de desenvolvimento do SNSAnalyser, foram utilizadas técnicas de projeto e análise baseadas no paradigma de orientação a objetos (resumidamente chamado de OO). A opção pela elaboração de um projeto OO se deu com a finalidade de se obter, ao final da implementação, um sistema que reflita os requisitos funcionais de forma condizente. Essa escolha também se baseou no fato do paradigma orientado a objetos permitir a reutilização de *software*, proporcionando não apenas a possibilidade de criação de novos componentes a partir do desenvolvimento do SNSAnalyser, como também o reuso de componentes já existentes.

No apêndice A deste documento estão relacionados alguns diagramas elaborados durante o desenvolvimento da ferramenta SNSAnalyser, segundo a notação UML™ (OMG, 2008). Entretanto, alguns dos artefatos produzidos durante o processo de desenvolvimento do sistema são apresentados ao longo da próxima seção com o objetivo de melhor descrever o seu modelo de implementação.

Além do paradigma da orientação a objetos, o processo de desenvolvimento da ferramenta SNSAnalyser foi modelado seguindo alguns padrões de projeto. O uso de padrões de projeto ajuda a nomear, abstrair e identificar os aspectos-chave de uma estrutura de projeto

comum a fim de torná-la útil para a criação de um projeto orientado a objetos reutilizável (GAMMA, 2000).

No intuito de atender ao requisito de disponibilidade através da *web*, a ferramenta SNSAnalyser foi desenvolvida na linguagem de programação Visual C#™ (MICROSOFT, 2007). Essa linguagem fornece suporte à utilização de componentes, padrões de projeto e orientação a objetos. Além disso, é uma das linguagens que pode ser utilizada na plataforma ASP .NET™, a plataforma *web* da Microsoft. Em conjunto com a solução adotada, o gerenciador de banco de dados, utilizado pelo SNSAnalyser é o Microsoft SQL Server 2005 (MICROSOFT, 2007), uma vez que é uma opção robusta e escalável, além de possuir uma versão gratuita: o SQL Server 2005 Express (MICROSOFT, 2007).

Por fim, com o objetivo de agregar valor à ferramenta, possibilitando que outras máquinas sejam utilizadas como proxy e assim adotando mais uma estratégia anti-bloqueio, o SNSAnalyser permite que seja utilizada a infra-estrutura de Power Proxy e Power Manager. Além disso, no desenvolvimento do sistema, foi introduzida também a utilização do Power Spider e do Power Script. Todos esses recursos serão mais bem detalhados na seção 4.4.

4.3 PRINCIPAIS FUNCIONALIDADES

As funcionalidades do SNSAnalyser foram projetadas de acordo com os requisitos dispostos na solução apresentada no presente trabalho. Assim, faz-se necessário um maior detalhamento sobre o comportamento de cada funcionalidade da referida ferramenta, objeto das seções subseqüentes.

4.3.1 Manter rede social

A funcionalidade “Manter Rede Social” provê o cadastro principal necessário para permitir a extração de redes sociais a partir de algum site de relacionamento. Através dessa funcionalidade, o administrador de rede social pode manter o registro das redes que podem ser alvo de extração e de todo o aparato que é necessário para realizar esse processo.

Para cadastrar uma rede social, o usuário do sistema deve acessar essa funcionalidade através do *menu* Cadastros > Redes Sociais. A partir daí, uma tabela com as redes sociais incluídas é exibida e pode-se optar pela inclusão de uma nova rede ou alteração ou remoção

de uma já existente. A seqüência de atividades executada nessa funcionalidade encontra-se ilustrada no diagrama exibido na Figura 10.

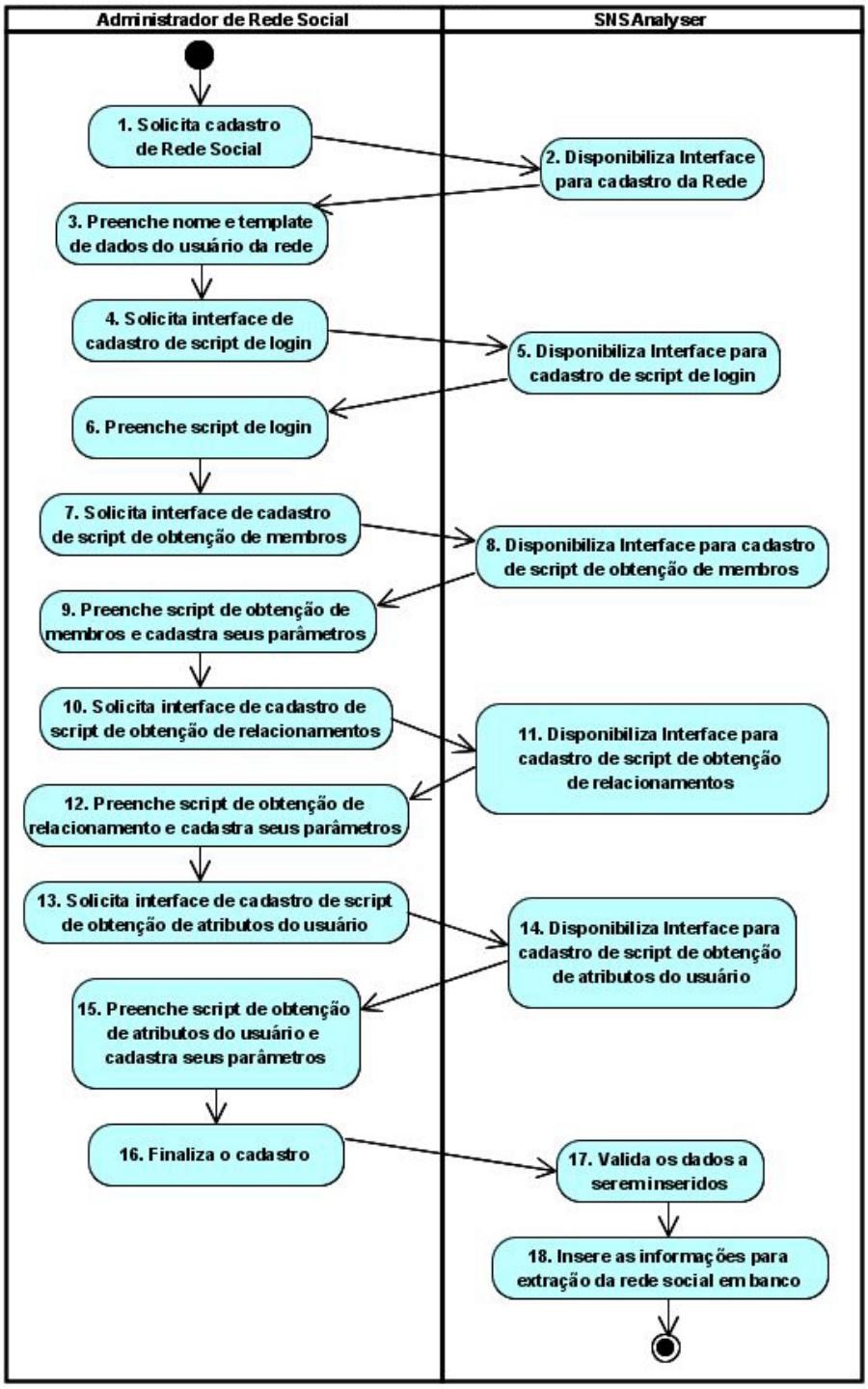


Figura 10 - Diagrama de Atividades – Manter Rede Social

Na opção de inclusão/alteração de rede social, uma *interface* com cinco abas é disponibilizada, possibilitando preenchimento do *template* daquela rede social e dos *scripts* de

log in, obtenção de membros de uma comunidade, obtenção de relacionamentos entre os membros extraídos e obtenção de atributos do usuário.

Após a inclusão de redes sociais, é necessário ainda passar por mais uma funcionalidade para então ser possível iniciar a extração de informações da rede social. Essa funcionalidade que também é crucial na viabilização da extração é o cadastro de usuários navegadores, que será abordado na seção seguinte.

4.3.2 Manter usuário navegador

Essa funcionalidade tem como objetivo associar usuários da rede social de cujos dados serão extraídos ao cadastro de uma rede social no SNSAnalyser. Desta forma, esses usuários serão utilizados para realizar a navegação no site de relacionamento, permitindo a obtenção de informações de lá.

Para evitar bloqueios por parte dos sites alvo, a ferramenta possibilita o cadastro de quantos usuários navegadores o administrador de rede social achar necessário. É importante ressaltar que a criação desses usuários deve ser feita de acordo com os termos de uso do site de relacionamento. Para os casos em que mais de um usuário navegador é associado a uma rede social, o SNSAnalyser utiliza aleatoriamente um deles, realizando, assim, uma distribuição aproximadamente equivalente de seu uso na navegação no site de relacionamento.

O acesso a essa funcionalidade se dá através do *menu* “Cadastros > Usuários Navegadores”, onde é apresentado um *combo* para seleção da rede social. Após a seleção, um grid exibe os usuários navegadores associados a essa rede e possibilita a alteração, exclusão e inserção de novos. As informações necessárias para o registro são o usuário e a senha utilizados na rede selecionada. Um diagrama de atividades relacionado a essa funcionalidade pode ser visto no apêndice A. Vale ressaltar que a senha é criptografada antes de armazenar em banco para maior segurança desses usuários.

4.3.3 Extrair rede social

A extração da rede social a partir dos sites de relacionamento é feita em duas etapas. Primeiro, são extraídos os membros de uma ou mais comunidades que irão compor os nós da rede. Depois, são extraídos os relacionamentos entre esses nós. Apesar de serem duas ações

distintas, a fim de prover uma visão mais ampla da operação, o diagrama de atividades abaixo relaciona toda a seqüência executada pelo pesquisador de rede social para realizar esse procedimento.

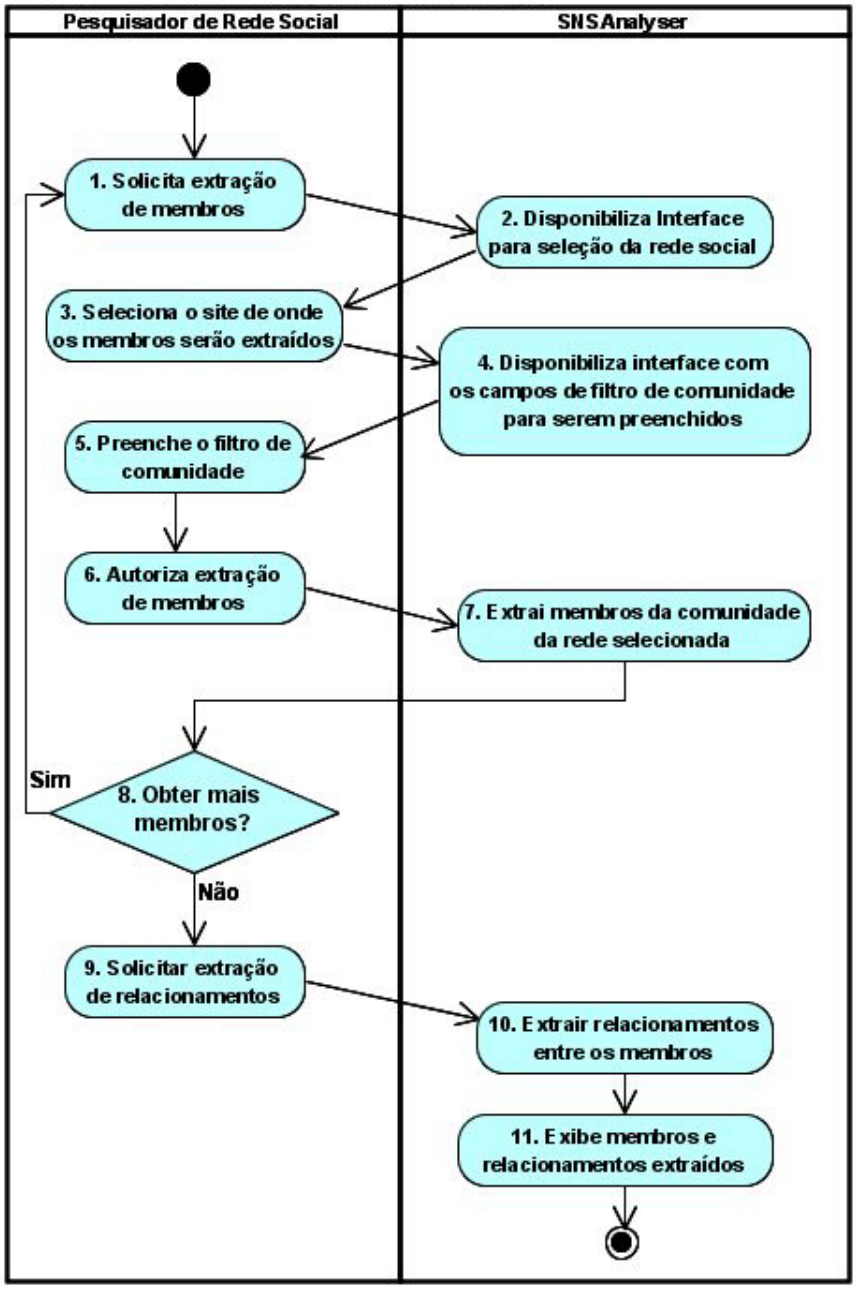


Figura 11 - Diagrama de Atividades – Extrair Rede Social

Como se pode notar, a extração de membros pode ser feita quantas vezes o pesquisador achar necessária, possibilitando colocar numa única rede social membros de mais de uma comunidade do site de relacionamento. Existe ainda uma extração complementar da rede social que não está apontada no diagrama acima. É a extração de atributos do usuário.

Essas informações somente são obtidas no momento em que o pesquisador de rede social solicita visualizá-las ou na exportação completa para Excel™.

Ao final da extração, o usuário é direcionado para uma tela com as informações de membros e relacionamentos extraídas do site de relacionamento. Essa funcionalidade pertence ao caso de uso “Manter dados extraídos” que é descrito a seguir.

4.3.4 Manter dados extraídos

Esse caso de uso visa, basicamente, prover funcionalidades de acesso aos dados extraídos. Uma vez que esse caso de uso lida com informações obtidas em outros sites de relacionamento, optou-se por não possibilitar a inclusão de informações a essa base de outra forma que não seja utilizando a funcionalidade de extração. Além disso, apenas são permitidas alterações e exclusões na base para ajustar a rede social em função de erros durante a extração ou preparar a base para realização de uma nova extração.

Nesse contexto, as funcionalidades contempladas nesse caso de uso são: Exibir membros, Exibir relacionamentos, Exibir membros e relacionamentos, Exibir atributos do usuário, Painel de Status e Apagar todos os dados extraídos. Excetuando-se a exibição de atributos do usuário, todas as outras funcionalidades podem ser acessadas através do *menu* “Dados Extraídos”.

Já para visualizar os atributos do usuário, basta estar em alguma das telas de exibição de dados extraídos ou de resultados de cálculos e clicar no identificador do usuário que fica na coluna Id, Id1 ou Id2, dependendo da *interface*.

A funcionalidade de exclusão dos dados extraídos, acessível através do *menu* “Dados Extraídos > Apagar Tudo”, exclui todos os registros de cálculo, relacionamentos e membros de comunidades. Desta forma, nenhum registro da rede social obtida permanece na base, o que possibilita novas extrações de redes sociais totalmente diferentes para serem analisadas.

Através do Painel de Status é possível visualizar quantos membros foram extraídos com sucesso, com erro, quantos faltam ter seus relacionamentos extraídos e quantos estão sendo utilizados para extração pela ferramenta naquele momento. Além disso, é possível retornar os membros com erro para a fila de extração ou remove-los da rede social que está sendo montada.

4.3.5 Manter métricas customizadas

A manutenção de métricas customizadas é uma funcionalidade que não afeta o fluxo básico de extração e análise de redes sociais, mas provê um poder muito grande ao pesquisador. Apesar do SNSAnalyser disponibilizar métricas pré-definidas, elas podem não atender em sua plenitude à análise que se deseja fazer, o que tornaria a avaliação insuficiente. Neste caso, a solução pode ser exportar a rede extraída para terminar a análise em uma outra ferramenta ou implementar a métrica necessária e incluí-la no SNSAnalyser.

Neste tópico, é abordado apenas o cadastro de métricas customizadas. Maiores detalhes sobre como realizar essa implementação, podem ser vistos na seção 5.3.

Para cadastrar uma métrica customizada no SNSAnalyser, deve-se acessar o *menu* “Cadastro > Métricas”. Em seguida, uma tabela é exibida com as informações de métricas já cadastradas. Nesta tela, pode escolher a inclusão de novas métricas ou optar pela remoção ou alteração de métricas já cadastradas.

As informações necessárias para o cadastro são o nome da métrica e o nome completo da classe, incluindo todo seu *namespace,* além da realização do *upload* da DLL com o código desenvolvido.

4.3.6 Calcular métricas

Dois tipos de métrica podem ser calculados no SNSAnalyser: as métricas pré-definidas, que são métricas embutidas no sistema e as métricas customizadas, que podem ser inseridas pelos usuários na ferramenta.

O cálculo de métricas pré-definidas está disponível através do *menu* “Métricas > Pré-definidas > Calcular”, enquanto o cálculo de métricas customizadas deve ser acessado pelo *menu* “Métricas > Customizadas > Calcular”. Caso todas as métricas customizadas inseridas na ferramenta implementem a interface IMetricFromNetwork, esse cálculo pode ser acionado antes do cálculo de métricas pré-definidas. Caso contrário, isso não deve ocorrer, uma vez que as métricas customizadas que implementam a interface IMetricFromDistance e IMetricFromResults dependem do cálculo de métricas pré-definidas.

Para ambos os cálculos, o processamento exige apenas um clique do pesquisador. Entretanto, nesse momento, o sistema executa algumas rotinas relativamente complexas. A

seqüência de procedimentos ocorrida no cálculo de métricas pré-definidas pode ser visto no diagrama da Figura 12.

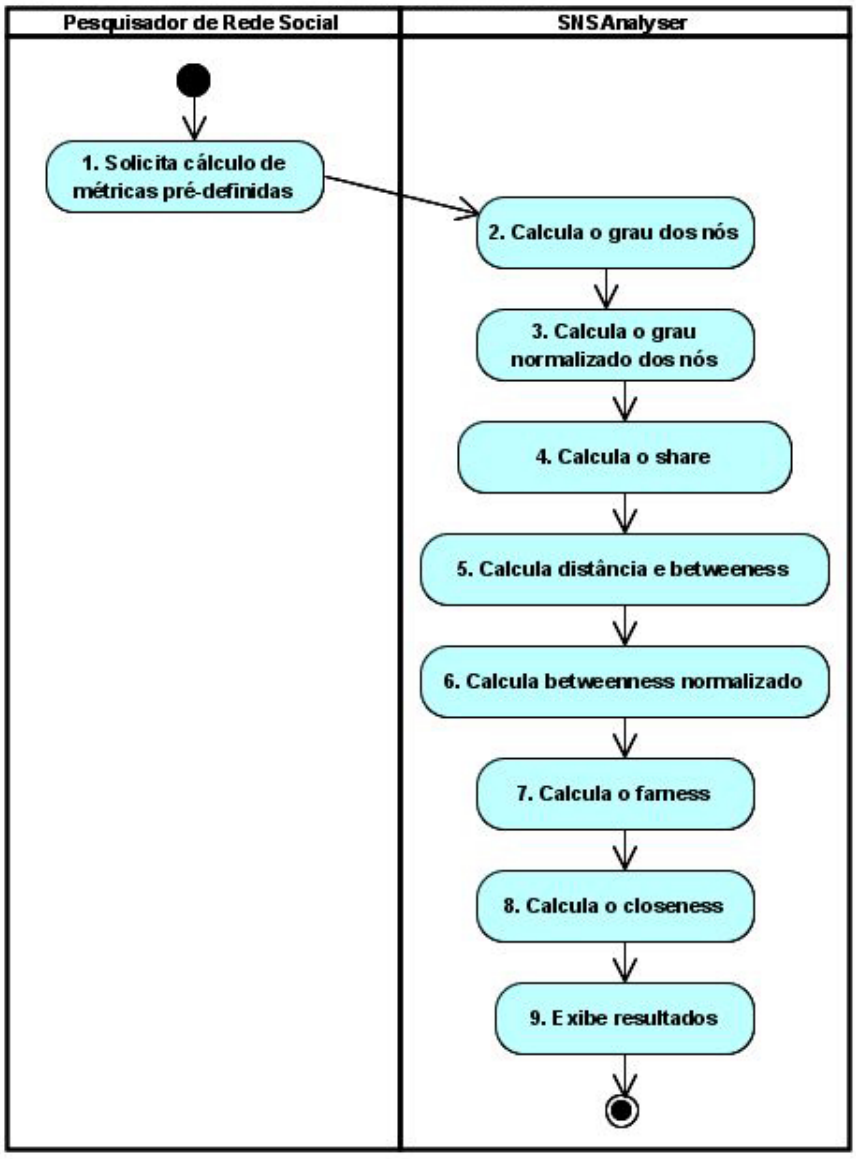


Figura 12 - Diagrama de Atividades – Calcular métricas pré-definidas

Conforme pode ser visto no diagrama, o cálculo de métricas pré-definidas engloba o cálculo dos graus e graus normalizados dos nós, *share*, distâncias entre os nós, *betweenness* e *betweenness* normalizado, *farness* e *closeness*. Abaixo, na Figura 13, segue outro diagrama de atividades. Desta vez, com os procedimentos realizados no cálculo de métricas customizadas.

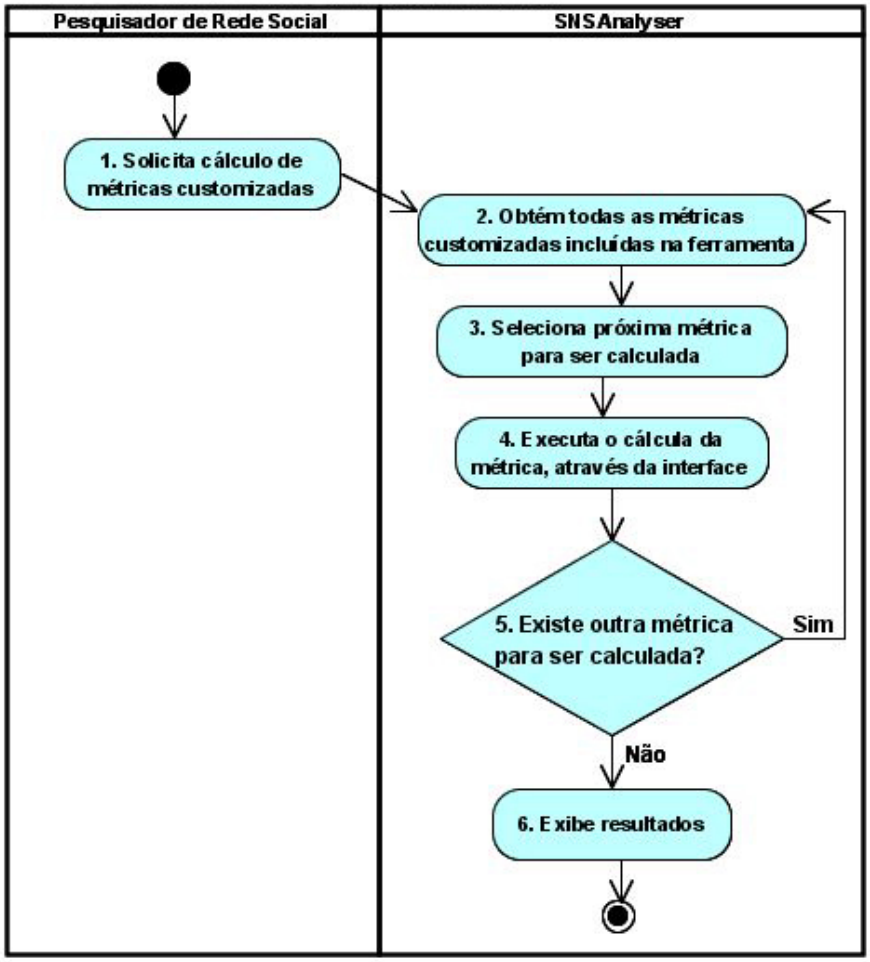


Figura 13 - Diagrama de Atividades – Calcular métricas customizadas

Tanto no caso das métricas pré-definidas, como no das customizadas, ao final do processamento, o usuário é direcionado para uma tela que mostra os resultados obtidos. A seção a seguir é dedicada à exibição dos resultados.

4.3.7 Exibir resultados dos cálculos

No caso de uso “Exibir resultados dos cálculos”, duas funcionalidades podem ser acessadas: a exibição de resultados do cálculo de métricas pré-definidas e a exibição do resultado do cálculo de métricas customizadas.

As duas funcionalidades são simples e consistem apenas na exibição das informações, sem maiores processamentos. Elas podem ser acessadas através do *menu* “Métricas > Pré-definidas > Exibir” e do *menu* “Métricas > Customizadas > Exibir” para pré-definidas e customizadas, respectivamente. Nestas telas, os resultados dos cálculos para cada membro da rede são exibidos e podem ser ordenados de acordo com qualquer métrica. Para facilitar a

visualização, as tabelas são paginadas, exibindo 10 registros por página e os atributos do usuário podem ser visualizados com apenas um clique em cima do seu identificador.

Para os casos em que as métricas pré-definidas e customizadas não suprem toda a necessidade da análise da rede social, pode-se optar pela exportação das redes extraídas, como é abordado a seguir na seção 4.3.8.

4.3.8 Exportar rede social

Três tipos de exportação são permitidas pelo SNSAnalyser. Duas delas são para Excel™, planilha eletrônica da Microsoft™, com a diferença que uma é mais simples e a outra mais completa. A terceira exportação é para o UCINet (BORGATTI; EVERETT; FREEMAN, 2002), uma das principais ferramentas disponíveis para análise de redes sociais, conforme foi visto no capítulo 2. Essa exportação serve para utilização na ferramenta UCINet como no NetMiner (NETMINER, 2008). Já a opção de exportação para Excel™ é em razão das limitações do formato reconhecido pelo UCINet, que não permite que os atributos do usuários sejam incluídos e também para prover interface com outras ferramentas.

A exportação para UCINet, acessível através do *menu* “Exportação > Ucinet - DL” gera um arquivo com os relacionamentos extraídos. Como existe uma limitação de 18 caracteres para o tamanho dos *labels* (BORGATTI; EVERETT; FREEMAN, 2002), optou-se por realizar essa exportação apenas com o identificador do nó da rede, já que outros atributos do usuário poderiam facilmente ultrapassar esse tamanho. O formato do arquivo exportado para o UCINet fica, então, conforme o quadro 5.

```
dl n=7 format = edgelist1
labels embedded
data:
id1 id2
id2 id1
id1 id3
id3 id1
id1 id4
id4 id1
id2 id5
id5 id2
id3 id6
id6 id3
```

Quadro 5 - Formato de arquivo exportado para o UCINet

Seguindo esse mesmo padrão, a exportação simples para Excel™ também só leva os relacionamentos entre os membros da rede e, para não ficar diferente, também utiliza os

identificadores dos nós para mapear as ligações entre os usuários. Entretanto, como o Excel não possui a mesma limitação do UCINet, além da exportação dos relacionamentos, também é exportada uma relação mapeando os identificadores dos relacionamentos, com o site de relacionamento em que foi obtido e seu identificador nesse site. Além disso, no Excel™, os relacionamentos estão mapeados em apenas uma direção, uma vez que é muito mais simples duplicar esses relacionamentos no sentido contrário do que removê-los dentro de uma lista que pode estar muito grande. Utilizou-se uma estratégia diferente nessa exportação uma vez que a exportação para UCINet é especifica para aquela ferramenta e procurou-se atender todas as imposições da ferramenta, enquanto a exportação para Excel™ objetiva atender a uma exportação para qualquer ferramenta. Desta forma, a exportação realizada através do *menu* "Exportação – Excel Simples" gera um arquivo com duas planilhas cada uma com uma tabela no formato exibido no quadro 6.

id1	id2
Id1	Id2
Id1	Id3
Id1	Id4
Id1	Id5
Id1	Id6
Id2	Id3
Id2	Id6
Id3	Id4
Id3	Id6
Id4	Id5
Id4	Id6
Id5	Id6

Id	IdUser	IdNetwork
Id1	413319963823171621	ORKUT
Id2	18283690547527467027	ORKUT
Id3	3335586048842373532	ORKUT
Id4	4162897139153130598	ORKUT
Id5	14960838652516644897	ORKUT
Id6	7883781898513781921	ORKUT

Quadro 6 - Formato de arquivo simples exportado para o Excel

Por fim, a exportação completa leva, além da tabela de relacionamentos, exibida a esquerda no quadro 6, uma segunda tabela mais completa que a segunda tabela da exportação simples. Assim, além dos três campos do formato simples, são levados também os atributos dos usuários, que são customizados e obtidos através de script, podendo variar de acordo com a rede social. Essa exportação é realizada acionando-se o item do *menu* "Exportação > Excel com Atributos".

4.4 INFRA-ESTRUTURA UTILIZADA

No desenvolvimento da ferramenta SNSAnalyser, foram utilizados alguns recursos de tecnologia pertencentes à Power Comunicação e Mídia Ltda (POWER.COM, 2008). Seu uso

e divulgação foi devidamente autorizado e seu funcionamento será esclarecido nas seções a seguir.

4.4.1 Power Spider

O Power Spider é um *web crawler* que interpreta Power Script. Ou seja, o Power Spider, recebe um XML (*Extensible Markup Language*), arquivo com linguagem de marcação conforme recomendação do W3C (W3C, 2008), com as instruções de navegação e manipulação de HTML interpreta e traduz em comandos de navegação na *web*. Ao final, as variáveis declaradas no script podem ser obtidas através de método específico do Power Spider.

A API (*Application Programming Interface*) do Power Spider é muito simples e fácil de ser utilizada. Existe uma classe chamada ProcessPageDocument que recebe, em seu construtor, o Power Script que será interpretado. Essa classe tem propriedades que definem seu método de conexão, com os servidores e portas a serem utilizados e métodos para atribuição de parâmetros, contexto e obtenção de variáveis de retorno. No Quadro 7, segue uma relação das principais propriedades e métodos do Power Spider com uma descrição de cada um.

Construtores	Descrição
ProcessPageDocument	Cria uma instância do spider, recebendo o script por parâmetro.
Propriedades	**Descrição**
ConnectMethod	Define o método de conexão do spider. Os valores permitidos são: Local, SocketProxy (Power Proxy) e ServerManager (Power Manager).
ConnectPort	Deve ser atribuída com a porta de conexão, em caso de conexão com Power Proxy ou Power Manager.
ConnectServer	Deve ser atribuída com o nome da máquina ou IP, em caso de conexão com Power Proxy ou Power Manager.
Métodos	**Descrição**
GetCookies	Obtém os cookies, que compõem o contexto, retornados pelo spider
GetVar	Obtém uma variável retornada pelo spider.
GetVars	Obtém todas as variáveis retornadas pelo spider.
Process	Executa o spider e, consequentemente, o Power Script
SetContextString	Atribui o contexto a ser utilizado.
SetVar	Passa um parâmetro pro spider.

Quadro 7 - Principais membros da classe ProcessPageDocument

No intuito de centralizar o acesso e simplificar a manutenção, foi criado um projeto chamado SNSAnalyser.SpiderExecuter que, compilado, gera uma DLL (*Dynamic Link Library*) que é referenciada pelo SNSAnalyser. Esse projeto possui quatro classes: ScriptContext, ScriptDocument, ScriptExecuter e ScriptParameter. Assim, ao invés da ferramenta acessar o *spider* diretamente, ela o faz através dessas classes, de forma que qualquer alteração nessas classes reflete em todo o sistema.

Após falar sobre o Power Spider, a seção seguinte relata o funcionamento do Power Script, peça fundamental para o funcionamento do *spider* e meio pelo qual se permitiu que o usuário da ferramenta utilizasse o *web crawler* de forma genérica, extraindo redes sociais de quaisquer sites de relacionamento que lhe interessasse.

4.4.2 Power Script

O Power Script é uma linguagem de programação de alto nível baseada no XML que possibilita a navegação e extração de dados aplicável a qualquer site da Internet. O Power Script é uma linguagem totalmente baseada em regras e variáveis de forma que novas regras podem ser adicionadas ao Power Script sem que, com isso, o programador da linguagem tenha que aprender novos conceitos, ficando praticamente intuitiva a adição dessa nova regra. Desta forma, aplica o princípio de projeto de sistema conhecido como programação extensível (ENGLISH, 2008), onde novas funcionalidades podem ser adicionadas sem prejudicar o código existente.

A finalidade básica do Power Script é extrair dados de qualquer documento HTML de forma relativamente simples para o programador. É possível, por exemplo, extrair o texto delimitado por identificadores, extrair um determinado atributo de um nó HTML, obter o item em foco num campo de seleção entre outras funcionalidades.

O Power Script é uma linguagem interpretada por um *spider*, que lê o conteúdo do script e executa a navegação e outros procedimentos conforme as regras passadas. Essa linguagem se baseia no XML e possui todas as principais estruturas existentes em uma linguagem de programação estruturada, como variáveis, comandos de decisão, repetição e tratamento de erros. Além das regras de navegação é possível adicionar outras regras ao *spider*, como manipulação de banco de dados, gravação de arquivos, envio de e-mails dentre outras. Para desenvolver um Power Script, o programador de scripts precisa saber conceitos básicos de lógica de programação e saber ler e interpretar documentos HTML.

Um script pronto pode ser parametrizado, logo pode ser utilizado com diferentes características em função dos parâmetros fornecidos. A execução do script permite a coleta de dados que podem ser facilmente recuperados por qualquer aplicação. A linguagem possui uma API simples que permite qualquer aplicação facilmente passar parâmetros ou recuperar os dados extraídos.

Apesar da execução do script efetuar uma navegação automatizada em sites da Internet a fim de extrair as informações de interesse, até o presente momento, não se tem notícia de um servidor ter conseguido detectar diferenças entre uma navegação humana e a navegação através do Power Script, desde que o programador do script tenha tomado os devidos cuidados conforme as estratégias apresentadas no presente documento.

4.4.2.1 Estrutura do Power Script

A estrutura do script é definida em duas partes: “context” e “rules”, conforme exemplo abaixo:

```
<processpage>
<context name="nome_do_contexto" />
<rules>
      <rule action="nome_da_acao">
            <!--
                  Outras ações
            -->
      </rule>
      <!--
            Outras ações
      -->
</rules>
</processpage>
```

Quadro 8 - Estrutura de um Power Script

O contexto são informações conseguidas através da navegação normal em um *website*, como por exemplo, *cookies* e indicadores de sessão. Um mesmo contexto pode ser compartilhado por diversos scripts. Portanto, se um script, por exemplo, efetuar a autenticação do usuário em um determinado site, o contexto resultante poderá ser aproveitado em outro script, que será interpretado como se o usuário já estivesse autenticado. O nome do contexto é definido no script de *log in* e tem a extensão .context.xml. A regra “context” define um nome para o contexto resultante da navegação.

Já na seção “rules”, pode-se colocar uma ou mais regras do tipo “rule” em cadeia ou aninhadas. Desta forma, o *spider* interpreta o Power Script, percorrendo-o em profundidade e, consequentemente, levando em consideração a seguinte ordem de processamento:

```
<processpage>
<rules>
      <rule action="primeira">
      </rule>
      <rule action="segunda">
            <rule action="terceira">
                  <rule action="quarta">
                  </rule>
            </rule>
            <rule action="quinta">
            </rule>
      </rule>
      <rule action="sexta">
      </rule>
</rules>
</processpage>
```

Quadro 9 - Ordem de processamento das regras de um Power Script

As regras que podem ser processadas dentro de um Power Script são diversas. No quadro abaixo, segue uma breve descrição do que cada uma delas executa.

Rules	Descrição
call	Realiza chamadas a procedures no banco de dados.
database	Interage com o banco de dados executando qualquer script DML.
delay	Provoca uma pausa na execução do script pelo tempo informado.
email	Envia e-mails.
executeif	Limita a execução de um determinado bloco do script para apenas os casos em que uma determinada condição é satisfeita.
for	Realiza uma iteração por um determinado número de vezes.
for-each	Percorre cada elemento de uma variável do tipo vetor e gerar uma repetição com esses valores.
read	Realiza requisições a páginas da *web* via GET ou POST.
savefile	Salva o conteúdo de uma URL ou o conteúdo de uma variável no sistema de arquivos
setvar	Possibilita ao usuário definir uma ou mais variáveis dentro do script.
throw	Lança uma exceção com uma mensagem específica.
xml	Permite a criação de variáveis do tipo XML.

Quadro 10 - Regras que podem ser utilizadas dentro de um Power Script

São muitos os atributos que podem ser utilizados junto com cada regra e, portanto, eles não serão abordados em detalhes nessa seção. No próximo capítulo, no exemplo de uso

da ferramenta, alguns exemplos de Power Scripts serão colocados. No apêndice B deste documento, é descrita uma API simplificada do Power Script.

4.4.3 Power Proxy

O objetivo do Power Proxy é possibilitar que a uma aplicação realize a requisição a um site através de outra máquina. Desta forma, sua utilização transfere a responsabilidade de realizar a requisição de um computador para outro.

O Power Proxy nada mais é do que um serviço que fica executando em uma máquina e que possui uma interface de comunicação bem definida com o Power Spider. A natureza do Power Proxy é passiva, ou seja, ele fica o tempo todo aguardando que um comando lhe seja enviado. Após a execução do comando, a conexão é encerrada imediatamente. Portanto não existem mecanismos de persistência da conexão. Através de suas propriedades o PowerSpider pode ser informado que não é para a máquina local fazer a requisição ao site da internet. Ao invés disso, o *spider* irá se conectar ao Power Proxy, utilizando uma conexão *socket* e ele é quem realizará a requisição a esse site. Feito isso, o *proxy* devolve o resultado para o *spider* e fecha a conexão, conforme é mostrado na Figura 14, cabendo ao Power Spider continuar o processamento a partir daí.

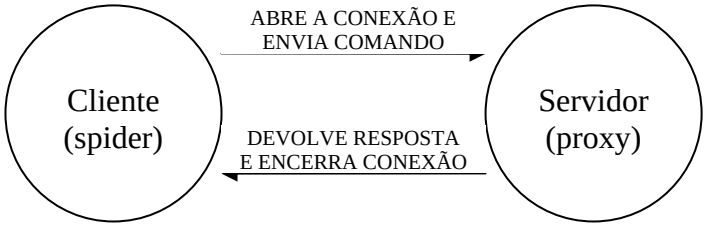


Figura 14 - Esquema de comunicação entre o spider e o proxy

A utilização de um *proxy* pelo *spider* é simples. Basta informar, nas propriedades do Power Spider, que o ConnectMethod utilizado será SocketProxy e atribuir ao ConnectServer e ao ConnectPort o IP do servidor e a porta de comunicação, respectivamente.

Entretanto, o uso de apenas um *proxy* não soluciona o problema em caso de bloqueio do IP por parte dos servidores do site de relacionamento. Para isso, pode-se utilizar de forma complementar outro serviço associado ao Power Proxy, o Power Manager. Ele tem a habilidade de gerenciar, controlar e acompanhar a situação de vários Power Proxies,

retornando para o *spider* aquele que está mais apto a atender alguma solicitação naquele momento. Uma breve explanação sobre o Power Manager é feita na seção 4.4.4.

4.4.4 Power Manager

O Power Manager é um serviço que interage com uma lista de Power Proxies e serve de interface entre eles e uma aplicação cliente. O papel do Power Manager é possibilitar a utilização de vários Power Proxies diferentes por parte de uma aplicação cliente. Neste caso, o sistema que está utilizando os serviços do Power Manager não precisa se preocupar com o gerenciamento dos Power Proxies, verificando qual deles está disponível ou tem menor fila no momento. Cabe ao Power Manager realizar essa avaliação e retornar para o cliente o Power Proxy mais apto a realizar a requisição solicitada naquele momento.

Através da utilização do Power Manager, a interação entre essas aplicações fica da seguinte forma:

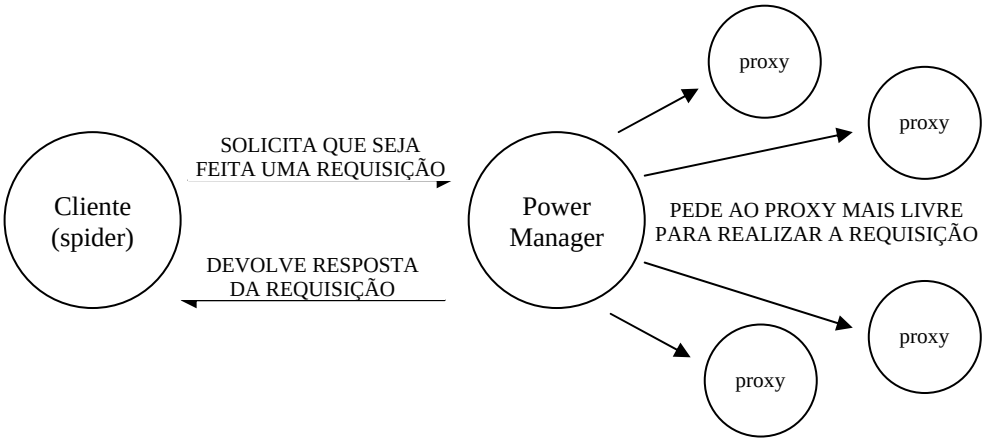


Figura 15 - Esquema de comunicação entre o spider e o manager

Vale ressaltar que essa estrutura de Proxy e Manager é opcional no SNSAnalyser, devendo ser utilizada caso exista a possibilidade de bloqueio de IP por parte do site de relacionamento. Caso não haja essa necessidade, a utilização de requisições locais é mais aconselhada por apresentar desempenho maior.

Uma vez apresentada a estrutura utilizada pelo SNSAnalyser, é o momento de detalhar a arquitetura da ferramenta e sua estrutura de implementação. A seção seguinte é destinada especificamente para isso.

4.5 ARQUITETURA DO SISTEMA

O SNSAnalyser segue uma arquitetura baseada em camadas, dividindo-se em três: camada de interface, camada de negócio e camada de acesso a dados. A adoção do padrão de projeto Facade (GAMMA, 2000) cria uma quarta camada que atua entre a camada de negócio e a de interface para algumas funcionalidades ou pode ser referenciada pela camada de negócios em outros casos. Para complementar essa divisão em camadas, foi utilizado o padrão de projeto DTO (*Data Transfer Object*), descrito por Fowler (2003), onde são criados objetos que transportam os dados entre as camadas.

Apoiando essa divisão em camadas, o SNSAnalyser foi modularizado em três bibliotecas e um *website*. A figura a seguir representa graficamente a arquitetura da ferramenta SNSAnalyser.

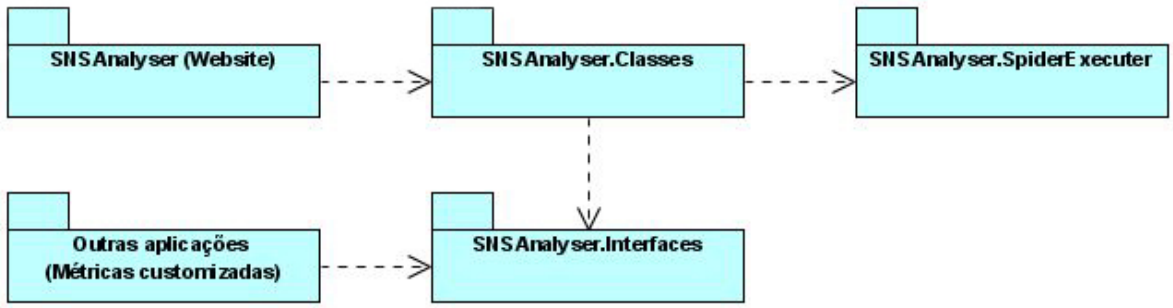


Figura 16 - Arquitetura com interdependência entre os módulos da ferramenta SNSAnalyser

O *website* SNSAnalyser corresponde à camada de interface visual do sistema, que pode ser colocada disponível na *web*.

A biblioteca SNSAnalyser.SpiderExecuter foi criada para ser o ponto central de acesso ao *spider*, consequentemente a interface entre o SNSAnalyser e a estrutura da Power.

A biblioteca SNSAnalyser.Interfaces foi criada para ser referenciado como interface pelas métricas customizadas que forem implementadas para serem atreladas à ferramenta.

A biblioteca SNSAnalyser.Classes é a responsável por toda a lógica de negócio e acesso a dados do SNSAnalyser, Ele é o coração do sistema e foi subdividido em 5 pacotes nas quais as classes estão inseridas de acordo com suas responsabilidades. Abaixo, é apresentada uma relação dos pacotes e uma descrição do seu conteúdo.

- BLL: *Business Logic Layer*. Neste pacote se encontram todas as classes que contêm as lógicas de negócio da aplicação.

- DAL: *Data Access Layer*. É onde se encontram as classes de acesso a dados da aplicação. Nesta camada, todas as classes foram implementadas de acordo com o padrão de projeto Singleton (GAMMA, 2000), de forma que apenas uma instância dessas classes é criada.

- DTO: *Data Transfer Object*. São nesse pacote que estão disponíveis as classes responsáveis pelo transporte de informações entre as camadas.

- Facade: Neste pacote estão disponíveis quatro classes: Calculator, Extractor, Exporter e Cryptography. Cada uma dessas classes é responsável por facilitar a utilização das funcionalidades de cálculo de métricas, extração de redes sociais, exportação das redes sociais extraídas e criptografia de senhas, respectivamente.

- DataSets: Analogamente ao uso de DTOs, em algumas situações, optou-se pelo uso de DataSets tipados, estruturas inerentes ao *framework* .NET da Microsoft™ que também podem ser usadas pra transportar dados entre as camadas. Na seção 4.5.4 são relacionadas os DataSets implementados e o motivo pelo qual seu uso foi preferido em detrimento do DTO.

Para melhor entendimento das atuações dos componentes desses pacotes, a Figura 17 mostra como essas camadas interagem entre si.

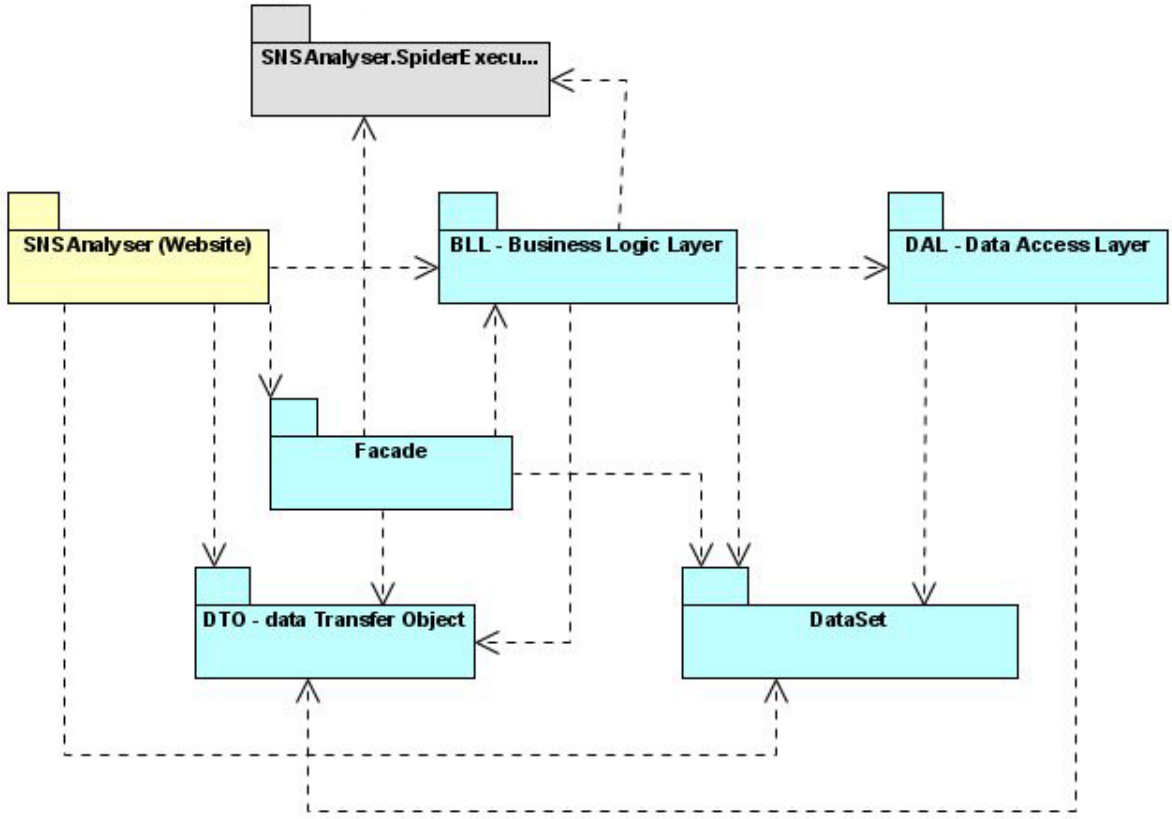


Figura 17 - Interação entre os módulos detalhando as camadas do assembly SNSAnalyser.Classes

Uma vez abordada a arquitetura da ferramenta, a seção seguinte explica como foram modelados os bancos de dados do SNSAnalyser e a partir daí é feita uma explicação mais detalhada de como é composto cada módulo do sistema.

4.5.1 Banco de dados

A primeira decisão na modelagem do banco de dados é: em quantos bancos devem estar armazenados os dados da aplicação? Comumente é utilizado apenas um banco de dados para armazenar as informações da aplicação inteira, mas esta é a melhor opção? No caso do SNSAnalyser, a resposta é não.

Nitidamente a aplicação tem dados de naturezas bastante distintas e com variações de volume muito grandes. Por um lado, uma porção dos dados que precisa ser armazenada se refere ao cadastro dos sites de relacionamento de onde serão extraídas as redes sociais a serem analisadas. De outro, os dados extraídos e os cálculos feitos em cima dessas informações também precisam ser guardados.

Enquanto as informações para extração são menos alteradas e tendem a ter um volume muito pequeno, os dados extraídos de outros sites juntamente com os cálculos realizados em cima deles tendem a ser muito mais voláteis e mais volumosos. Além disso, as informações extraídas e os cálculos realizados exigem maior processamento e disponibilidade do banco de dados que as informações para extração.

Neste contexto, a melhor opção para a ferramenta é isolar os dados de cadastro para extração de redes sociais e colocar em outro banco os dados para análise por parte do usuário. Assim feito, as figuras 18 e 19, apresentam o DER (Diagrama Entidade-Relacionamento) das duas bases.

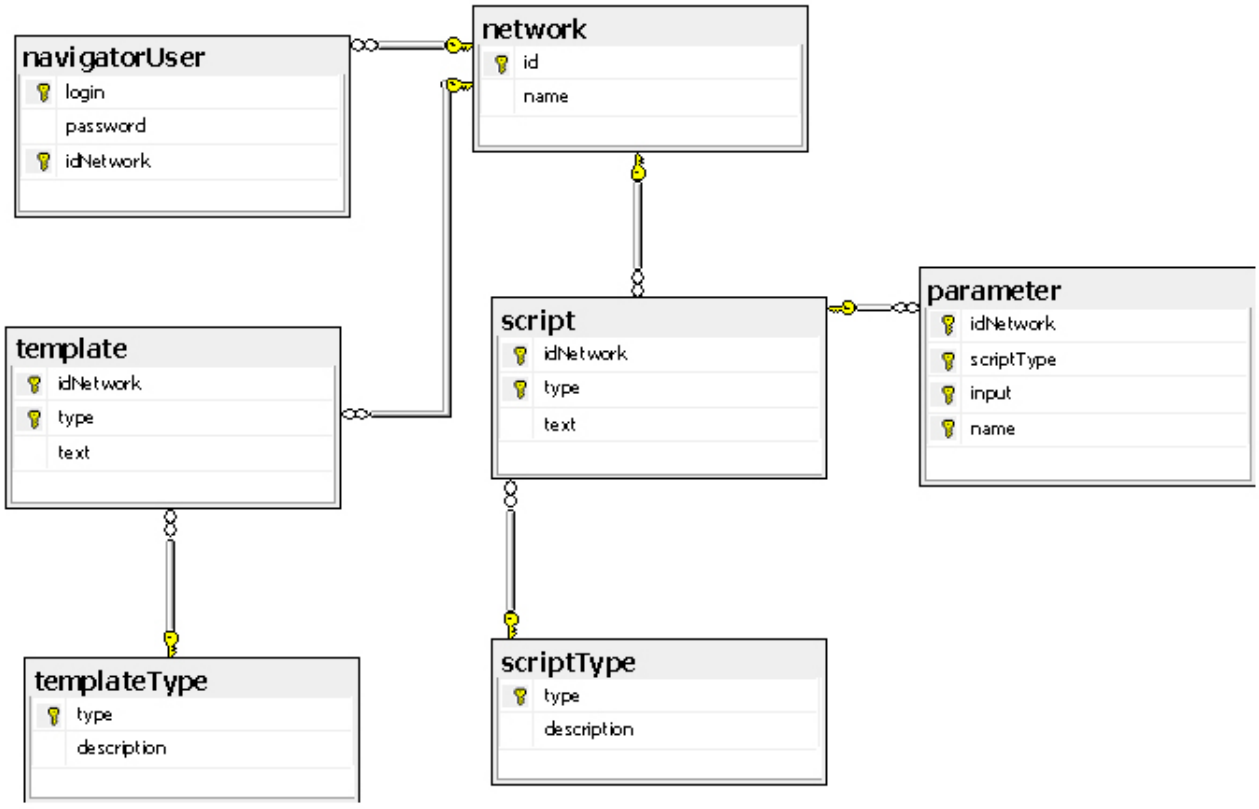


Figura 18 - DER - bdSNSAnalyser – Informações para extração

A tabela “network” do banco bdSNSAnalyser armazena aos nomes das redes sociais cadastradas para extração e cria um identificador para elas. Vinculado ao cadastro da rede, está o cadastro de quatro registros na tabela de script, cada um com um Power Script associado a uma ação diferente de acordo com os tipos existentes na tabela “scriptType”.

Na versão atual da ferramenta, a tabela “scriptType” contém quatro registros pré-cadastrados e essenciais para o funcionamento correto do sistema. São eles: 1-login, 2-

getMember, 3-getFriends e 4-getUserAttributes. Neste banco, existe ainda mais uma tabela de domínio, a “templateType” que possui o registro 1-showUserAttributes.

Apesar do modelo de dados estar preparado para a existência de mais de um *template* para cada rede social, a versão atual da ferramenta permite apenas a inserção de um registro na tabela “template”.

Associados ao cadastro de cada script podem ser inseridos quantos parâmetros forem necessários de entrada ou saída na tabela “parameter”, porém a aplicação realiza validações que impõem certas regras que precisam ser respeitadas nesse cadastramento.

A seguir são listados os tipos dos scripts e as regras associadas a eles:

- login: nenhum parâmetro de entrada e nenhum de saída. Neste caso não é permitido o cadastro de parâmetros, mas duas entradas são necessárias ao script: o usuário e a senha. Como saída não há parâmetros, apenas o contexto de autenticação.

- getMember: n parâmetros de entrada, mas somente um de saída, pois podem ser necessárias mais de uma informação para selecionar a comunidade, mas a saída sempre deve ser uma coleção de identificadores dos membros no site de relacionamento.

- getFriends: apenas um parâmetro de entrada e um de saída, onde o parâmetro de entrada é o identificador do usuário no site de relacionamento e o parâmetro de saída, uma coleção de identificadores dos usuários com os quais ele se relaciona.

- getUserAttributes: um parâmetro de entrada e *n* de saída para, cuja entrada deve ser o identificador do usuário no site de relacionamento e a saída pode ser qualquer atributo do usuário que seja interessante para o analisador da rede social resultante.

Por fim, na tabela “navigatorUser”, o usuário pode inserir quantos usuários navegadores ele achar que são necessários para cada rede, no intuito de evitar bloqueios a um usuário específico. Quando utilizados em pouca quantidade, a depender da rede social, eles podem ser bloqueados ou até mesmo excluídos sob a pena de terem realizado mais requisições do que as esperadas de um ser humano. Vale ressaltar que é importante que a criação desses usuários tenha sido feita de acordo com os termos de uso do site de relacionamento.

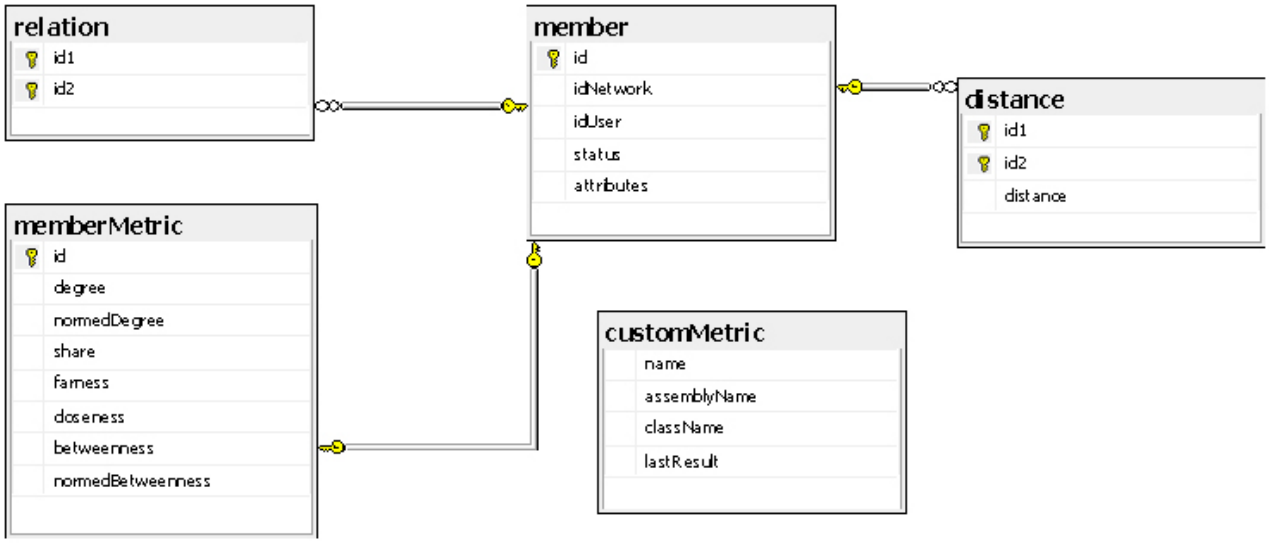


Figura 19 - DER - bdSNSNetworkData – Informações extraídas e relacionadas a cálculo

Já no banco de dados bdSNSNetworkData, são armazenadas as informações extraídas dos sites de relacionamento e toda a parte relacionada com o cálculo de métricas. A tabela “member” armazena os membros de uma ou mais comunidades de um site de relacionamento. A tabela “relation” registra os relacionamentos entre os membros obtidos.

Já com relação ao cálculo, a tabela customMetric armazena as medidas customizadas que podem ser implementadas pelos usuários e cadastradas para serem utilizadas pelo SNSAnalyser. A tabela “distance”, guarda as distâncias entre os nós da rede e a tabela “memberMetric” armazena o resultado dos cálculos pré-definidos realizados pela aplicação.

A seguir, cada módulo da aplicação é analisado separadamente.

4.5.2 SNSAnalyser.SpiderExecuter

O módulo SNSAnalyser.SpiderExecuter foi criado no intuito de garantir que todo acesso ao *spider* ficasse centralizado em um único componente. Para isso foram criadas quatro classes, representadas no diagrama da Figura 20.

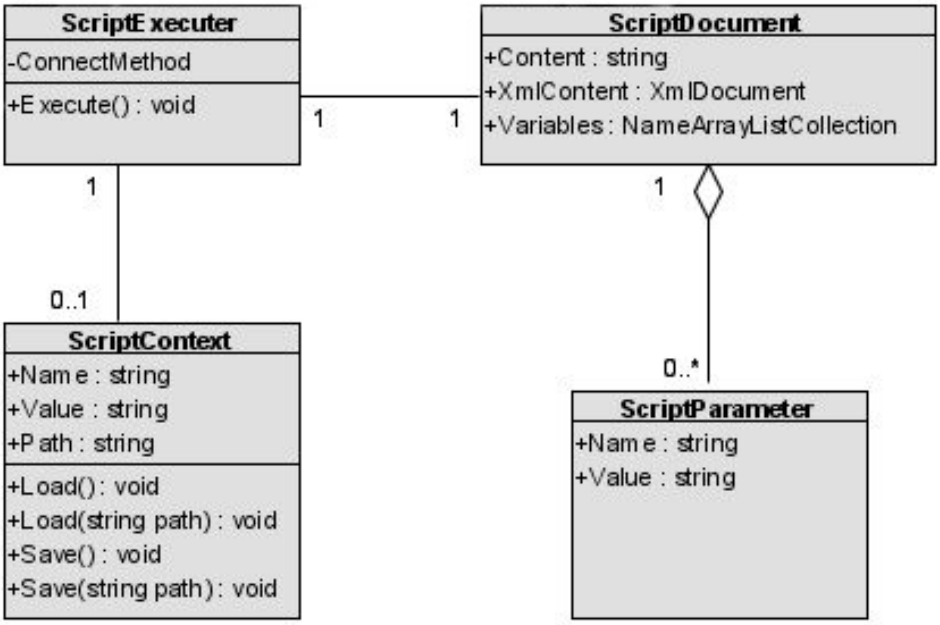


Figura 20 - Diagrama de Classes - SNSAnalyser.SpiderExecuter

A seguir, é feita uma breve descrição de cada uma dessas classes.

- ScriptParameter: Classe que representa um parâmetro do script.

- ScriptDocument: Classe que representa um script propriamente dito, com seus parâmetros e variáveis de retorno. Para facilitar seu uso, o conteúdo do script pode ser retornado por duas propriedades diferentes, onde uma delas retorna em formato *string* e a outra em XmlDocument, uma classe pertencente ao *framework* .NET™.

- ScriptContext: Classe que representa um contexto. Tem a habilidade de salvar e recuperar o contexto a partir de arquivos em disco.

- ScriptExecuter: Classe que tem a habilidade de executar um script, podendo associa-lo a um contexto ou não. Essa classe também é a responsável pela configuração do *spider* para utilização de Power Proxy ou Power Manager, caso se apliquem.

4.5.3 SNSAnalyser.Interfaces

O SNSAnalyser.Interfaces é o módulo responsável por disponibilizar as interfaces que devem ser implementadas pelas métricas customizadas pelos usuários. Esse *assembly* foi feito para ser distribuído aos usuários que tiverem esse objetivo e deve ser referenciado no projeto deles.

O SNSAnalyser.Interfaces possui três *interfaces:* IMetricFromNetwork, IMetricFromDistance, IMetricFromResults. Qualquer uma dessas interfaces pode ser implementada pelos usuários. A diferença entre elas está nos dados que elas recebem por parâmetro do SNSAnalyser.

A interface IMetricFromNetwork requer a implementação do método calculateFromNetwork que recebe por parâmetro uma string no formato XML com os relacionamentos entre os membros da rede social. O Quadro 11 mostra a estrutura da string em formato XML passada por parâmetro para o método que implementa essa *interface*.

```
<dsRelation xmlns="http://tempuri.org/dsRelation.xsd">
  <Relation>
    <id1>ID_1</id1>
    <id2>ID_2</id2>
  </Relation>
  <Relation>
    <id1>ID_2</id1>
    <id2>ID_4</id2>
  </Relation>
</dsRelation>
```

Quadro 11 - Estrutura do parâmetro do método calculateFromNetwork da interface IMetricFromNetwork

Caso o desenvolvimento da métrica necessite de mais informações do que simplesmente a rede de relacionamento, ele pode ser feita utilizando a *interface* IMetricFromDistance, que requer a implementação do método calculateFromDistance. Esse método recebe por parâmetro duas strings no formato XML. A primeira, da mesma forma que a interface anterior, contém os relacionamentos entre os membros da rede social e a segunda, as distâncias entre todos os nós da rede. O quadro 12 mostra a estrutura da segunda string em formato XML passada por parâmetro para o método que implementa essa *interface*.

```
<dsDistance xmlns="http://tempuri.org/dsDistance.xsd">
  <Distance>
    <id1>ID_1</id1>
    <id2>ID_2</id2>
    <distance>DISTANCIA_ENTRE_ID_1_E_ID_2</distance>
  </Distance>
  <Distance>
    <id1>ID_2</id1>
    <id2>ID_4</id2>
    <distance>DISTANCIA_ENTRE_ID_2_E_ID_4</distance>
  </Distance>
</dsDistance>
```

Quadro 12 - Estrutura do segundo parâmetro do método calculateFromDistance da interface IMetricFromDistance

Se a métrica customizada necessitar do resultado dos cálculos efetuados pelo SNSAnalyser, então a interface adequada é a IMetricFromResults. Ela exige a implementação do método calculateFromResults que recebe por parâmetro as duas strings das interfaces anteriores e mais uma, também em formato XML, com os resultados dos cálulos de métricas pré-definidas para cada membro da rede. O quadro 13 mostra a estrutura do último parâmetro passado para o método que implementa essa *interface*.

```
<dsMemberMetric xmlns="http://tempuri.org/dsMemberMetric.xsd">
  <MemberMetric>
    <id>ID_1</id>
    <degree>DEGREE_DO_ID_1</degree>
    <normedDegree>DEGREE_NORMALIZADO_DO_ID_1</normedDegree>
    <share>SHARE_DO_ID_1</share>
    <farness>FARNESS_DO_ID_1</farness>
    <closeness>CLOSENESS_DO_ID_1</closeness>
    <betweenness>BETWEENNESS_DO_ID_1</betweenness>
    <normedBetweenness>BETWEENNESS_NORM_DO_ID_1</normedBetweenness>
  </MemberMetric>
  <MemberMetric>
    <id>ID_2</id>
    <degree>DEGREE_DO_ID_2</degree>
    <normedDegree>DEGREE_NORMALIZADO_DO_ID_2</normedDegree>
    <share>SHARE_DO_ID_2</share>
    <farness>FARNESS_DO_ID_2</farness>
    <closeness>CLOSENESS_DO_ID_2</closeness>
    <betweenness>BETWEENNESS_DO_ID_2</betweenness>
    <normedBetweenness>BETWEENNESS_NORM_DO_ID_2</normedBetweenness>
  </MemberMetric>
</dsMemberMetric>
```

Quadro 13 - Estrutura do terceiro parâmetro do método calculateFromResults da interface IMetricFromResults

Por fim, todos esses métodos devem retornar uma string também em formato XML com o resultado obtido pelos seus cálculos. No quadro 14, é demonstrada uma estrutura similar à que deve ser utilizada nas strings retornadas nos três caso

```
<tabela_no_pai>
  <registro_no_que_se_repete>
    <coluna1>VALOR_DA_COLUNA_1_DO_REGISTRO_1</coluna1>
    <coluna2>VALOR_DA_COLUNA_2_DO_REGISTRO_1</coluna2>
    <coluna3>VALOR_DA_COLUNA_3_DO_REGISTRO_1</coluna3>
  </registro_no_que_se_repete>
  <registro_no_que_se_repete>
    <coluna1>VALOR_DA_COLUNA_1_DO_REGISTRO_2</coluna1>
    <coluna2>VALOR_DA_COLUNA_2_DO_REGISTRO_2</coluna2>
    <coluna3>VALOR_DA_COLUNA_3_DO_REGISTRO_2</coluna3>
  </registro_no_que_se_repete>
</tabela_no_pai>
```

Quadro 14 - Estrutura de retorno dos métodos das interfaces

A seguir, serão abordadas as classes centrais da ferramenta SNSAnalyser.

4.5.4 SNSAnalyser.Classes

Conforme foi visto anteriormente, o *assembly* SNSAnalyser.Classes foi subdividido em cinco *namespaces*.

As classes constituintes do *namespace* DTO e DataSets são apenas containeres para transportes dos dados pertencentes aos objetos. No *namespace* DTO, sete classes podem ser encontradas. Relacionadas ao cadastro para extração de informações estão as classes NavigatorUser, Network, Parameter, Script e Template. Já as classes CustomMetric e Member, estão vinculadas aos dados extraídos e ao uso de métricas. Cada uma dessas classes está destinada ao transporte e armazenamento em memória dos dados referentes ao objeto de mesmo nome.

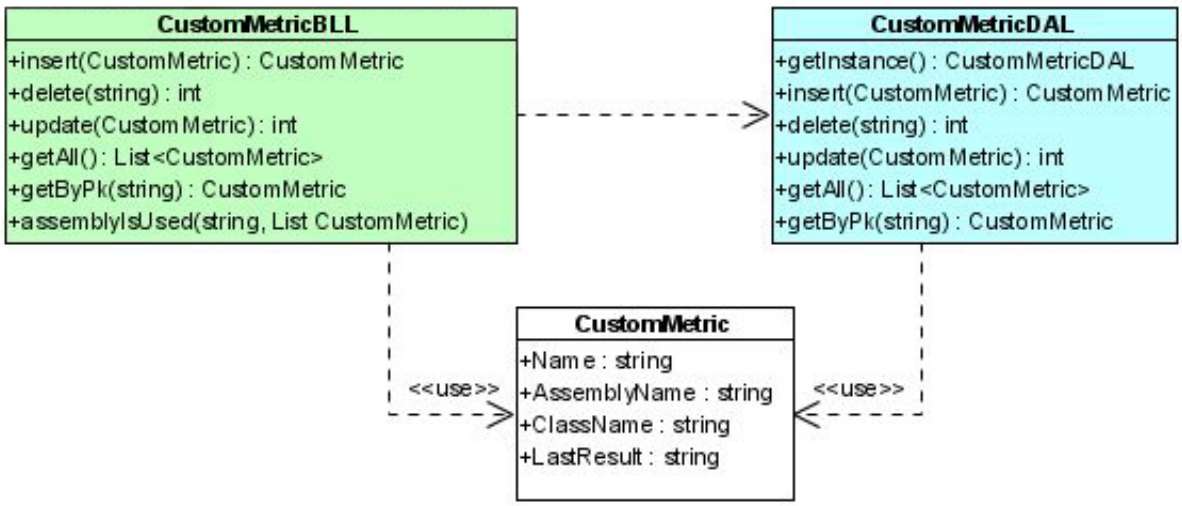


Figura 21 - Diagrama de Classes – CustomMetricBLL, CustomMetricDAL e CustomMetric

A Figura 21 mostra o diagrama de classes referente ao cadastro de métricas customizadas, utilizando o DTO para transportar informações entre as camadas.

O *namespace* DataSets é composto por quatro *DataSets* tipados: dsAttributes, utilizado na obtenção de atributos do usuário, dsDistance, dsRelation e dsMemberMetric, utilizados na obtenção de distâncias, relacionamentos e métricas pré-definidas. A opção pelo uso de *DataSets* nesses casos foi pelo fato serem necessárias conversões desses dados para XML, recurso já oferecido nos *DataSets*, mas que precisaria ser implementado nos DTOs. A

Figura 22 mostra a utilização de três dessas classes pelo *facade* Calculator, que passa para as métricas customizadas o conteúdo desses *DataSets* em formato XML.

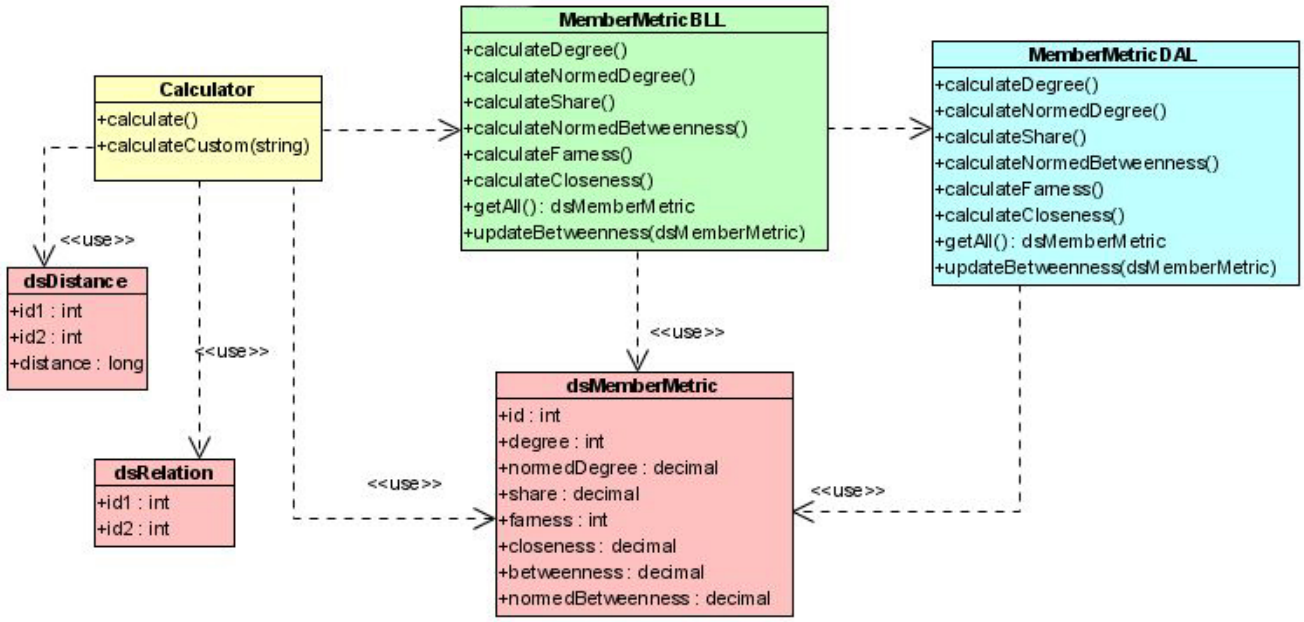


Figura 22 - Diagrama de Classes – Calculator e suas interdependências

As classes do *namespace* DAL são responsáveis pela interface da aplicação com o banco de dados. São elas que enviam comandos SQL (*Structure Query Language*) que realizam operações no banco de dados. Neste pacote, implementando o padrão de projeto Singleton (GAMMA, 2000), encontram-se as classes CustomMetricDAL, DistanceDAL, MemberDAL, MemberMetricDAL, RelationDAL, NavigatorUserDAL, NetworkDAL, ParameterDAL, ScriptDAL e TemplateDAL, das quais as cinco primeiras interagem com o banco de dados bdSNSNetworkData e as cinco últimas, com a base bdSNSAnalyser que armazena as informações para realização das extrações.

Já para as regras de negócios, foram desenvolvidas sete classes no *namespace* BLL. São elas: CustomMetricBLL, DistanceBLL, MemberBLL, MemberMetricBLL, NavigatorUserBLL, NetworkBLL, ParameterBLL, RelationBLL, ScriptBLL e TemplateBLL. Essas classes servem de ligação entre a interface do sistema e as classes de acesso a dados, além de realizarem validações e implementarem regras de negócio. A Figura 23 apresenta o diagrama de classes que envolve o cadastro de uma Network com seus *scripts*, parâmetros e *templates*.

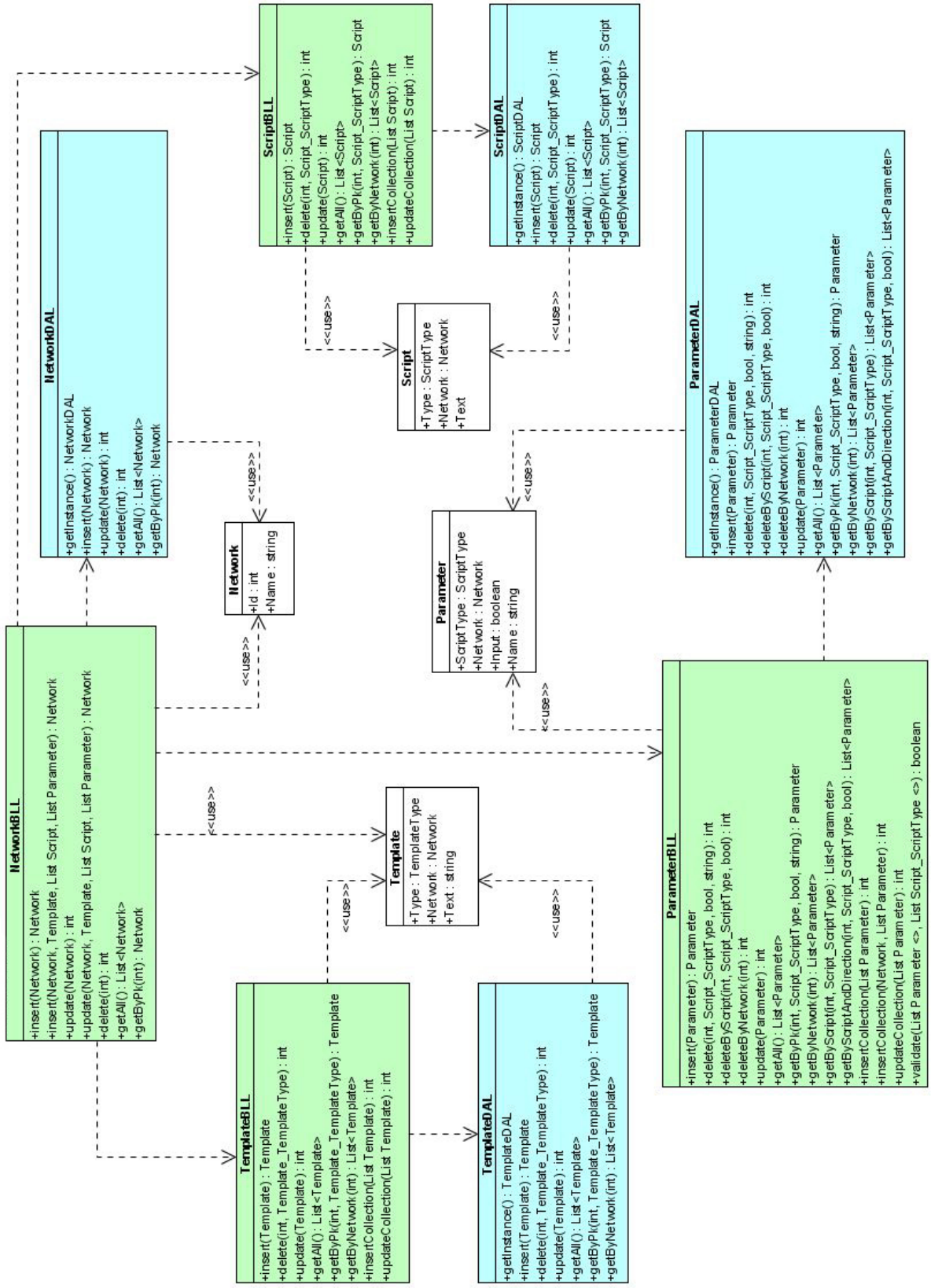


Figura 23 - Diagrama de Classes – Network, Script, Parameter e Template

Por fim, o *namespace* Facade é composto por quatro classes já citadas anteriormente.

A classe Extractor tem por objetivo prover uma interface única para todo tipo de extração de informações de outros sites.

A classe Calculator, cujo diagrama foi exibido na Figura 22, aciona tanto os cálculos pré-definidos no SNSAnalyser quanto cálculos de métricas customizadas.

Cryptography é a classe responsável pela criptografia e descriptografia de senhas.

A classe Exporter é quem provê o mecanismo de exportação de dados para outras ferramentas.

A Figura 24 mostra toda a complexidade da nuvem que o *facade* Extractor esconde.

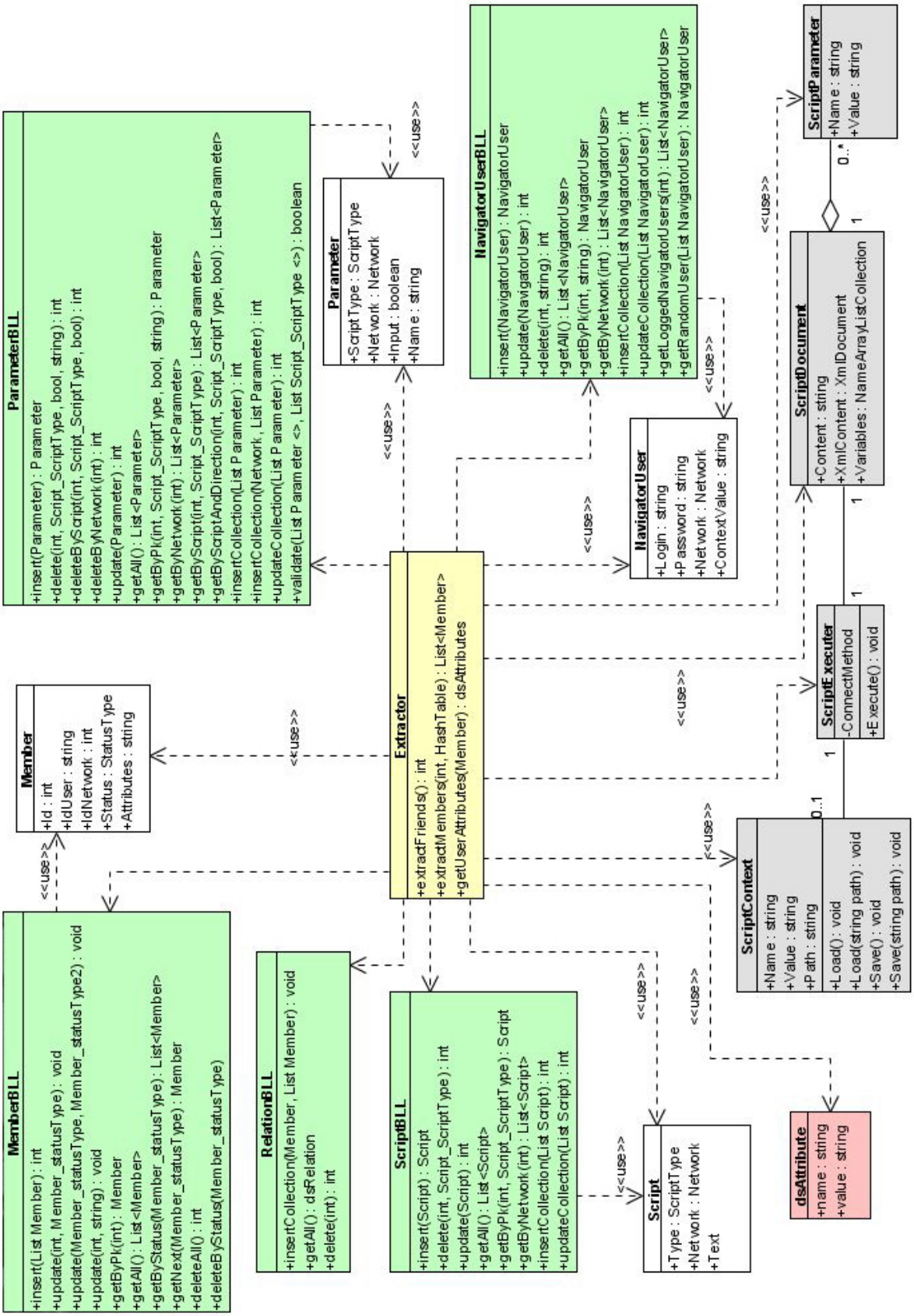


Figura 24 - Diagrama de Classes – Facade Extractor

No apêndice A, podem ser vistos outros diagramas de classes elaborados no processo de construção da ferramenta SNSAnalyser.

4.5.5 SNSAnalyser Website

O website SNSAnalyser é a interface com o usuário. Ele é composto de 11 telas que possibilitam a interação entre os usuários do sistema e a ferramenta. Para tanto, a ferramenta SNSAnalyser foi projetada de forma a prover uma interface amigável para os usuários, atentando para os aspectos visuais, como uma melhor disposição dos campos na tela.

Nesse contexto, é responsabilidade das classes pertinentes ao *website* prover interfaces de cadastramento para as redes sociais e todo o aparato necessário para a extração. Além disso, esse pacote também comporta as telas que permitem ao usuário extrair as redes, executar e visualizar os cálculos, incorporar novas métricas e exportar as informações para serem usadas em outros sistemas.

Dois papéis são identificados como utilizadores da ferramenta SNSAnalyser. Estes papéis podem ou não ser executados pela mesma pessoa. O primeiro deles, é o papel de administrador de redes sociais. Ele é o responsável por manter o cadastro de redes sociais, com seus *templates*, *scripts* e parâmetros e também manter o cadastro de usuários navegadores para as redes sociais cadastradas. A figura abaixo mostra o diagrama que contempla os casos de uso associados a esse ator.

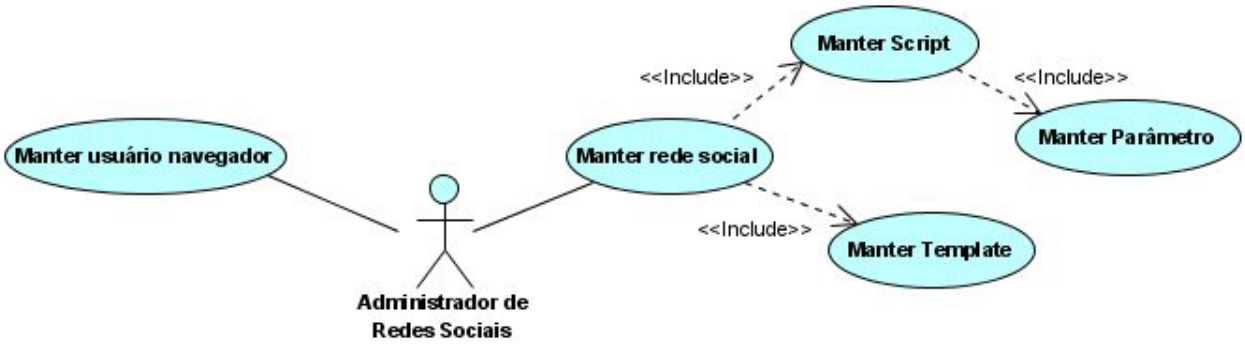


Figura 25 - Diagrama de Caso de Uso – Administrador de Redes Sociais

Além dos casos de uso apresentados na Figura 25, existem outros casos de uso que se referem às funcionalidades utilizadas pelo pesquisador de rede social, cuja atribuição consiste em fazer uso da ferramenta para extrair e analisar redes sociais extraídas de sites de

relacionamento. O Diagrama de Caso de Uso apresentado na Figura 26 referencia todas as demais funcionalidades disponibilizadas no SNSAnalyser.

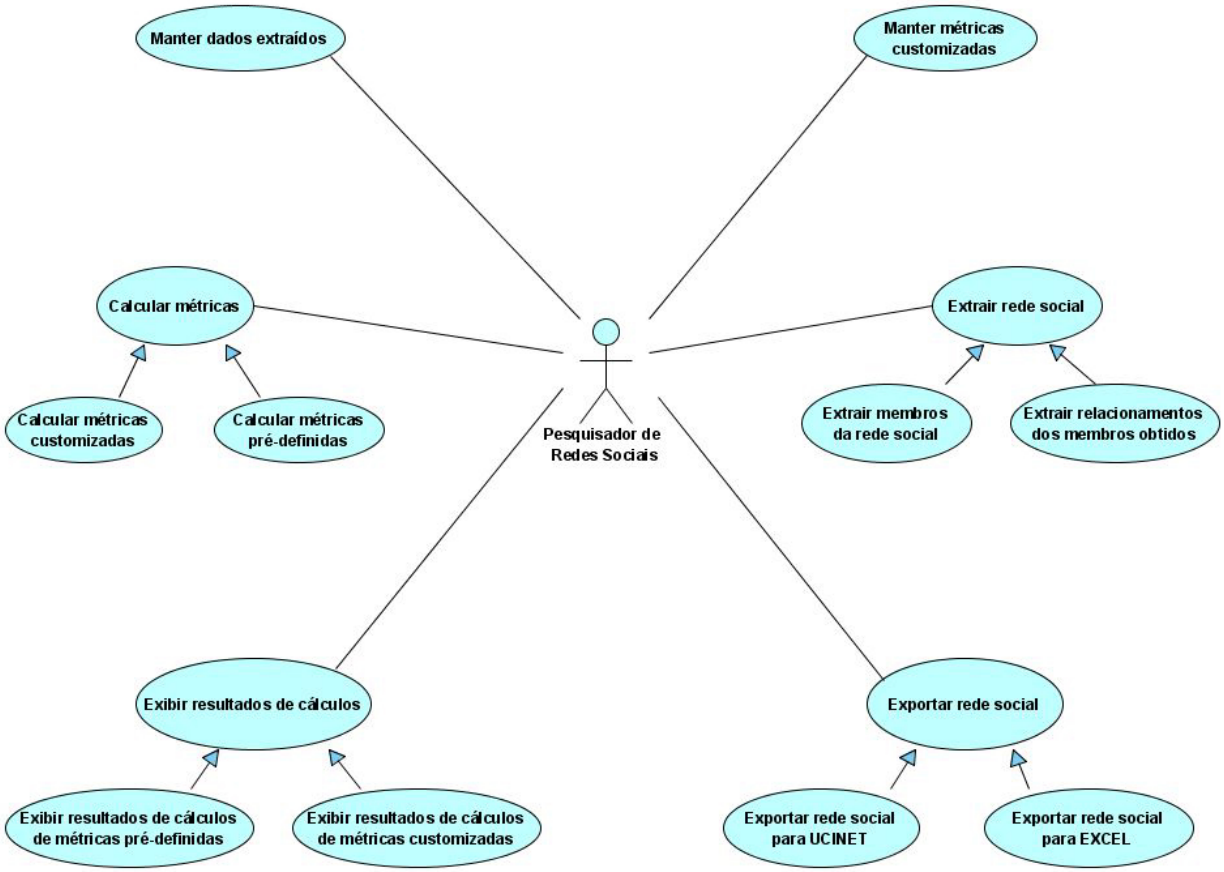


Figura 26 - Diagrama de Caso de Uso – Pesquisador de Redes Sociais

Analisando as figuras 25 e 26, é possível constatar que a ferramenta segue os requisitos direcionados pela solução proposta na presente pesquisa. Destaca-se ainda que as funcionalidades relacionadas ao administrador de rede social podem ser executadas pelo pesquisador, desde que ele conheça os princípios básicos do desenvolvimento de Power Script.

4.6 CONSIDERAÇÕES FINAIS

Neste capítulo discutiu-se a implementação da ferramenta SNSAnalyser, originada a partir das idéias apresentadas nesta dissertação. Sua principal contribuição reside em apoiar os pesquisadores de redes sociais na coleta e análise de redes sociais a partir de comunidades existentes em sites de relacionamentos, além de auxiliar profissionais interessados em utilizar redes sociais para promover produtos ou serviços de interesse de um determinado nicho de

mercado. Com este raciocínio, foram mostrados requisitos, características técnicas, funcionais, bem como alguns dos principais artefatos produzidos durante o processo de desenvolvimento, visando o melhor entendimento de como funciona a ferramenta.

Destaca-se que o SNSAnalyser foi construído à luz da solução proposta na presente pesquisa para extração e análise de redes sociais a partir de comunidades existentes em sites de relacionamento. A fundamentação teórica para a construção dessa ferramenta pode ser encontrada nos Capítulos 2 e 3.

No próximo capítulo será apresentado um exemplo de uso da ferramenta com o intuito de subsidiar a avaliação da aplicabilidade do SNSAnalyser na extração e análise de redes sociais.

5 EXEMPLO DE USO

Este capítulo visa exemplificar o uso do SNSAnalyser e efetuar uma avaliação sobre a solução proposta na presente pesquisa, analisando a aplicabilidade da ferramenta na extração e análise de redes sociais a partir de comunidades existentes em sites de relacionamento. Assim, o principal objetivo desse capítulo consiste em avaliar a capacidade da ferramenta de realizar esse processo, possibilitando a análise de redes sociais obtidas a partir de fontes antes inexploradas.

Para mostrar o funcionamento do SNSAnalyser, será apresentado o cadastramento da rede social Orkut™ com alguns usuários navegadores e uma métrica customizada. Em seguida, será feita a extração de uma rede social de interesse de uma empresa de artigos esportivos que planeja promover produtos de um nicho específico. O público desta campanha simulada são as pessoas praticantes do esporte esqui aquático e a avaliação será feita utilizando esse foco. Serão exibidos os resultados dos cálculos de métricas pré-definidas no sistema e da métrica customizada incorporada durante este exemplo na ferramenta. Ao final, será apresentada a exportação dos dados como forma de continuar a análise da rede social em outras ferramentas, pois essa funcionalidade pode ser utilizada nos casos em que as métricas aplicadas não apresentem resultados que atendam em sua totalidade ao interesse da análise. Como forma de documentação, os dados inseridos em cada situação e os resultados encontrados em cada cenário foram registrados.

Ao final, é feita uma avaliação sobre a ferramenta no que tange ao auxílio de empresas e profissionais que querem selecionar os usuários com um determinado perfil que se destacam dentro da rede social, constituindo-se em público-alvo bastante adequado para campanhas de promoção ou direcionamento de produtos.

O presente capítulo se inicia explicando como configurar a ferramenta para a extração de uma rede social para, em seguida, mostrar a extração dos dados e os resultados dos cálculos. Ao final, é feita uma avaliação geral da ferramenta SNSAnalyser, apontando seus pontos fortes e fracos.

5.1 CADASTRO DE REDE SOCIAL

Para o exemplo mostrado, utilizou-se o site de relacionamento Orkut™ como base para extração da rede social. Assim, tornou-se necessário o cadastramento dessa rede social no SNSAnalyser, incluindo seus scripts de login, obtenção de membros de comunidade, obtenção de relacionamentos entre as pessoas e obtenção de atributos do usuário, além do *template* para exibição dessas informações.

Para iniciar o procedimento, após selecionar no *menu* o item “Cadastros” e depois “Redes Sociais”, deve-se acionar o botão “Adicionar Rede”, conforme Figura 27.

Figura 27 - Tela de Seleção de Redes Sociais

Em seguida é apresentada a tela de cadastro de redes sociais com a primeira aba selecionada. Nela foi preenchido o nome da rede e o *template* para exibição dos atributos do usuário.

O *template* para exibição das informações do usuário nada mais é que um trecho de HTML onde são incluídas variáveis demarcadas com o sinal # que serão substituídas pelos parâmetros de saída do script de obtenção de atributos do usuário. Para tanto, eles devem estar com essa sincronia, a fim de que os resultados do script possam ser inseridos no *template*. A Figura 28 mostra a tela preenchida com o nome da rede social e o *template* utilizado.

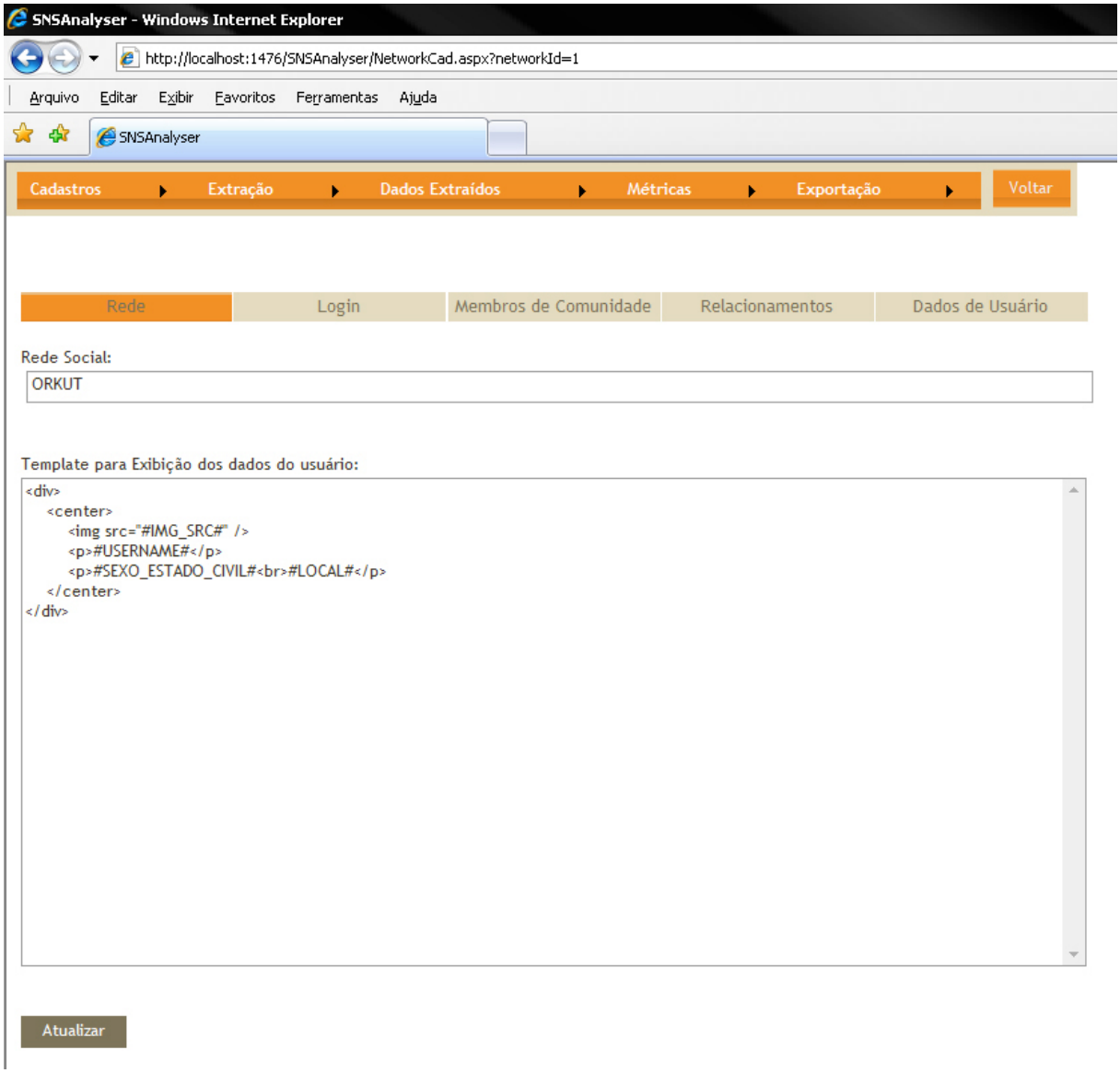


Figura 28 - Cadastro de redes sociais – Script de obtenção de dados do usuário

Para combinar com o *template* exibido na Figura 28, o script de obtenção de atributos do usuário precisa ter quatro parâmetros de saída e os nomes desses parâmetros devem ser iguais aos nomes das variáveis do template, ou seja, neste caso, IMG_SRC, USERNAME, SEXO_ESTADO_CIVIL e LOCAL. A quantidade de parâmetros e os nomes utilizados poderiam ser diferentes, desde que houvesse equivalência entre o script e o *template*. Portanto, fica a critério do pesquisador, obter os atributos do usuário e exibi-los da forma que melhor lhe convenha, levando apenas em consideração que essas informações precisam estar disponíveis no site de origem.

Já o parâmetro de entrada do script de obtenção de atributos não tem toda essa flexibilidade, a ferramenta impõe que seja apenas um atributo, que deve ser o identificador do

usuário naquela rede social. Entretanto, o nome desse atributo também é flexível e cabe apenas ao desenvolvedor do script escolher a denominação a ser utilizada. A Figura 29 mostra o cadastro do script de obtenção de atributos com seus parâmetros.

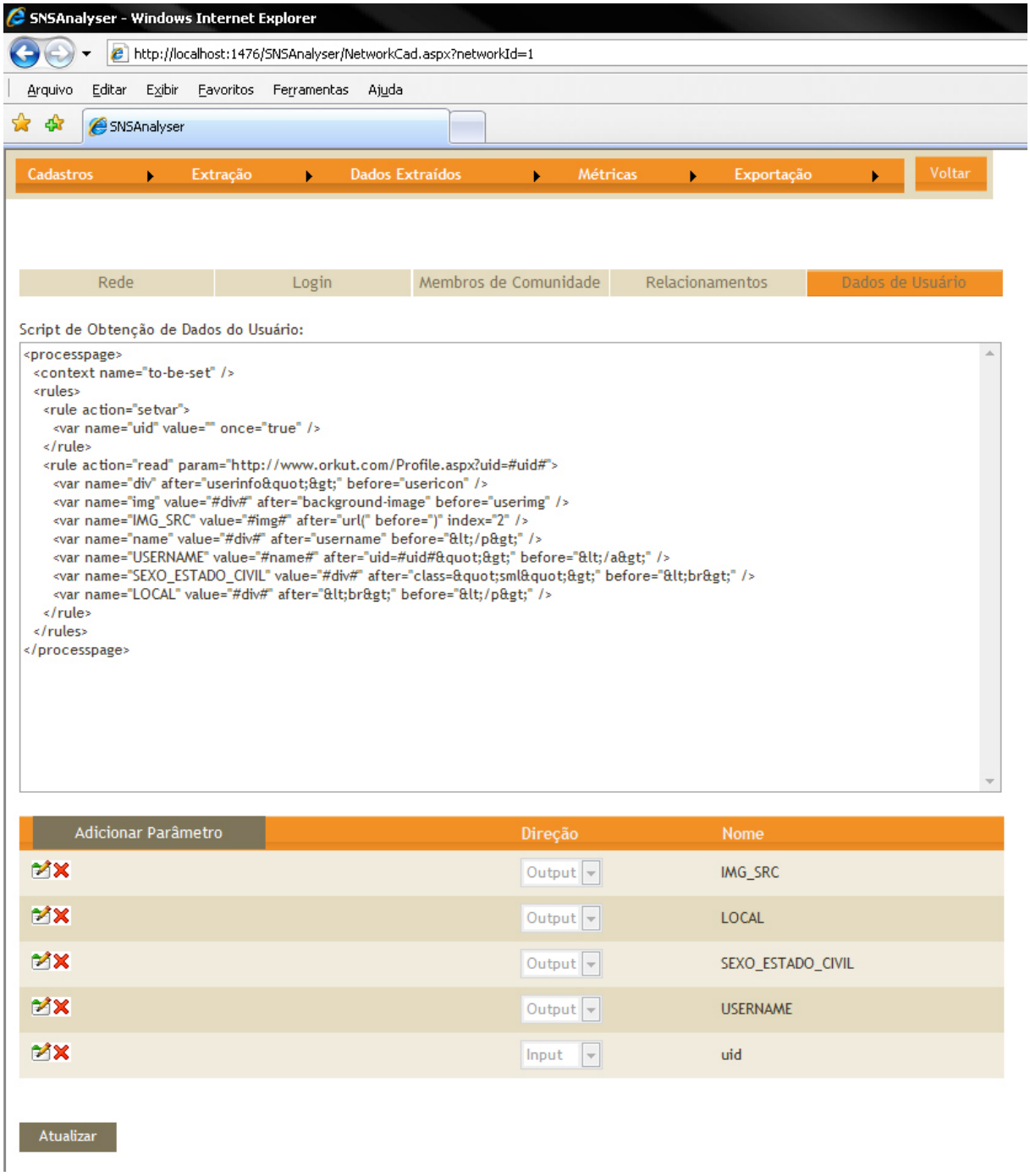


Figura 29 - Cadastro de redes sociais – Script de obtenção de dados do usuário

Ao analisar o script existente na Figura 29, pode-se perceber que ele realiza uma requisição na página do perfil do usuário passado como parâmetro (http://www.orkut.com/Profile.aspx?uid=#uid#) e, em seguida, efetua operações para retirar

da página retornada o conteúdo existente entre determinados trechos de HTML, atribuindo, dessa forma, o conteúdo aos parâmetros de saída do script.

O script de login é o responsável por autenticar os usuários navegadores na rede social para que eles possam acessar as páginas da *web* e extrair as informações dos sites de relacionamento. Por imposição da ferramenta, esse script não tem flexibilidade nenhuma de parâmetros, tanto que no cadastro não é permitido cadastrar nenhum parâmetro, nem de entrada nem de saída. Apesar disso, dois parâmetros de entrada são necessários para o seu funcionamento correto, são eles: username e password, usuário e senha, respectivamente. Desta forma, a implementação desse script deve prever a passagem desses parâmetros e seu resultado não deve gerar nenhum parâmetro de saída, apenas o contexto de autenticação para o *spider*. O Quadro 15, mostra o script de login utilizado no cadastro da rede social Orkut™.

```
<processpage>
  <context name="to-be-set" />
  <rules>
    <!-- define as variaveis de entrada -->
    <rule action="setvar">
      <!-- variaveis utilizadas no script -->
      <var name="language" value="pt-BR" once="true" />
      <var name="username" value="" once="true" />
      <var name="password" value="" once="true" />
    </rule>
    <rule action="read"
param="https://www.google.com/accounts/ServiceLoginAuth?service=orkut" delay="100">
      <post name="service" value="orkut" />
      <post name="hl" value="pt-BR_BR" />
      <post name="continue" value="http://www.orkut.com/RedirLogin.aspx?msg=0" />
      <post name="Email" value="#username#" />
      <post name="Passwd" value="#password#" />
      <post name="rmShown" value="1" />
      <post name="signIn" value="Sign in" />
      <var name="urlfinalizelogin" after="content=&amp;quot;0; url='" before="'&amp;quot;"
fix-amp="true" />
      <!-- O usuario nao confirmou o email no orkut -->
      <rule action="executeif" type="empty" cond="#urlfinalizelogin#">
        <rule action="setvar">
          <var name="urlfinalizelogin"
value="https://www.google.com/accounts/ServiceLogin?continue=http%3A%2F%2Fwww.orkut
.com%2FRedirLogin.aspx%3Fmsg%3D0&amp;service=orkut&amp;passive=true&amp;skipvpage=t
rue&amp;sendvemail=false" />
        </rule>
      </rule>
      <!--FIM  O usuario nao confirmou o email no orkut -->
      <!-- Regras de validacoes -->
      <validation not-exists="playCaptcha" code-erro="5004" />
      <validation value="#urlfinalizelogin#" not-exists="forgotpasswd" code-
erro="5002" />
      <validation value="#urlfinalizelogin#" not-exists="pagaerror" code-
erro="5003" />
      <validation value="#urlfinalizelogin#" not-exists="cmd=logout" code-
erro="5003" />
      <rule action="read" param="#urlfinalizelogin#">
        <var name="aux2" attribute="href" of-tag="a" index="1" />
        <var name="urlfinalizelogin2" after="0; url='" before="'" fix-amp="true"
index="1" />
        <!-- Regras de validacoes -->
```

```
        <validation not-exists="playCaptcha" code-erro="5004" />
        <validation value="#aux2#" not-exists="pagaerror" code-erro="5003" />
        <!-- Contempla a requisicao a SETID, se houver -->
        <rule action="read" param="#urlfinalizelogin2#">
          <var name="urlfinalizelogin3" value="#urlfinalizelogin2#:$:$"
after="continue=" before=":$:$" url-chars="decode" />
          <rule action="read" param="#urlfinalizelogin3#" />
        </rule>
      </rule>
    </rule>
    <catch code-erro-contains="Error to GET">
      <rule action="throw" message="5005" />
    </catch>
  </rules>
</processpage>
```

Quadro 15 - Script de Login do Orkut

A partir da análise do script de *login* acima, pode-se perceber que ele faz uma requisição para a URL do Google™ contida no atributo “param” da regra cuja ação é “read”, postando os parâmetros service, hl, continue, Email, Passwd, rmShown e SignIn. Após essa operação pode ser vista a utilização da regra de ação “executeif”, para verificar se o HTML retornado é de um usuário que não confirmou o e-mail no Orkut™ e algumas validações para garantir que a autenticação foi realizada com sucesso. Esses trechos do script estão destacados em negrito para facilitar a visualização.

Os scripts abordados são importantes para possibilitar o acesso ao site de relacionamento e obter informações a respeito dos usuários que fazem parte da rede social. Entretanto, os próximos dois scripts é que são os responsáveis pela extração e, consequentemente, formação da rede social. O primeiro deles, o script de obtenção de membros de comunidades, atua a partir de qualquer quantidade de parâmetros de entrada, mas requer que apenas um parâmetro de saída seja cadastrado. Cada parâmetro de entrada fará com que seja disponibilizado um campo com seu nome na interface de extração de membros. Já o parâmetro de saída deve conter um arranjo com os identificadores dos usuários naquele site de relacionamento.

Na Figura 30, pode-se observar que, no exemplo em questão, o script utiliza apenas um parâmetro de entrada, o identificador da comunidade, por isso, na extração de membros, apenas um campo, o cmmid, conforme nome dado ao parâmetro, é solicitado ao usuário. Pode-se notar também que, após a primeira requisição, é extraída, além dos membros que são exibidos na primeira página, a informação da quantidade de membros que participam daquela comunidade. Esse dado é importante, pois é necessário na iteração para trazer os membros que aparecem nas próximas páginas.

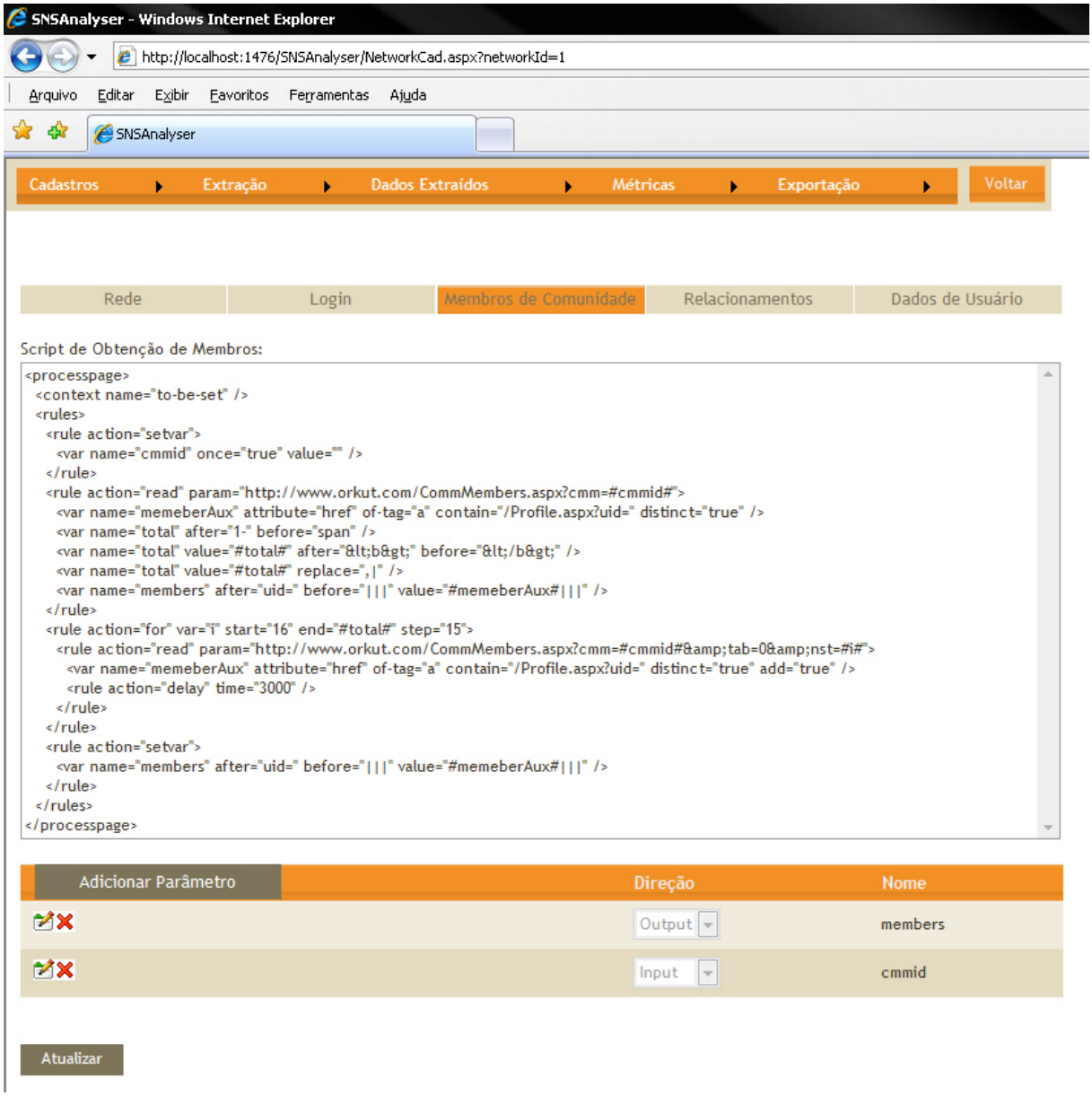


Figura 30 - Cadastro de redes sociais – Script de obtenção de membros de comunidade

Por fim, o script de obtenção de relacionamentos, mostrado no Quadro 16, conclui a formação da rede, através da realização da conexão entre os membros extraídos. Para isso, ele obtém todos os usuários que se relacionam com cada um dos membros no site de relacionamento e o sistema os associa a aqueles que constam como membros das comunidades extraídas. Para esse script é permitido apenas o cadastro de um parâmetro de entrada e um de saída, onde se sugere a utilização do identificador do usuário como entrada e o retorno de um *array* de identificadores de usuários aos quais ele se relaciona como saída.

```
<processpage>
  <context name="elmolcruz" />
  <rules>
    <rule action="setvar">
      <var name="id" once="true" value="not-set" />
    </rule>
    <rule action="read"
param="http://www.orkut.com/FriendsList.aspx?uid=#id#" valid-
exists="search" code-erro="Tela não pôde ser acessada.">
      <var name="html" />
      <var name="pno_aux" value="#html#" after="pno=" before="&quot;" />
      <rule action="executeif" cond="#pno_aux#" type="not-empty">
        <rule action="setvar">
          <var name="indice" value="#[lenarray(pno_aux)]#" />
          <var name="pno_end" value="#pno_aux#" index="#indice#" />
        </rule>
      </rule>
      <var name="aux" value="#html#" after="listitem" before="footer_r" />
      <rule action="executeif" cond="#aux#" type="not-empty">
        <rule action="setvar">
          <var name="friends" value="#aux#" after="/Profile.aspx?uid="
before="&quot;" distinct="true" />
        </rule>
      </rule>
      <rule action="executeif" cond="#aux#" type="empty">
        <rule action="setvar">
          <var name="friends" value="" />
        </rule>
      </rule>
    </rule>
    <rule action="for" var="i" start="2" end="#pno_end#">
      <rule action="read"
param="http://www.orkut.com/FriendsList.aspx?uid=#id#&amp;pno=#i#">
        <var name="out" />
        <var name="aux2" value="#out#" after="listitem" before="footer_r"
/>
        <var name="friends" value="#aux2#" after="/Profile.aspx?uid="
before="&quot;" distinct="true" add="true" />
        <rule action="delay" time="1100" />
      </rule>
    </rule>
  </rules>
</processpage>
```

Quadro 16 - Script de Obtenção de relacionamento (amigos) do Orkut

Nesse script, pode-se observar um exemplo de validação sendo feita no HTML retornado pela requisição. Uma vez que a página de amigos apresenta um formulário de nome “search”, ele foi utilizado para isso. Também pode ser vista a utilização de funções, como é o caso da “lenarray”, que retorna o tamanho do arranjo passado por parâmetro. Além disso, percebe-se que o script de obtenção de relacionamentos, assim como o script de obtenção de membros, também realiza uma iteração para obter as informações contidas nas páginas seguintes. Vale ressaltar que, em ambos os scripts, foram utilizados atrasos dentro dos laços, a

fim de evitar que o *spider* realize muitas requisições por segundo e, por conta disso, possa ser bloqueado pelo Orkut™.

5.2 CADASTRO DE USUÁRIOS NAVEGADORES

Para poder iniciar o processo de extração, ainda é necessário fazer mais um cadastro. Pelo menos um usuário navegador precisa ser cadastrado para cada rede, no intuito de permitir a navegação dentro dela.

Neste contexto, o procedimento utilizado é acessar o *menu* "Cadastros" e nele o item "Usuários Navegadores", selecionar a rede social na interface disponibilizada pelo sistema e, em seguida acionar o botão "Adicionar Usuário". Neste momento, devem ser preenchidos o login e senha do usuário navegador naquela rede social e solicitado o salvamento das informações. A Figura 31 mostra essa tela de cadastro.

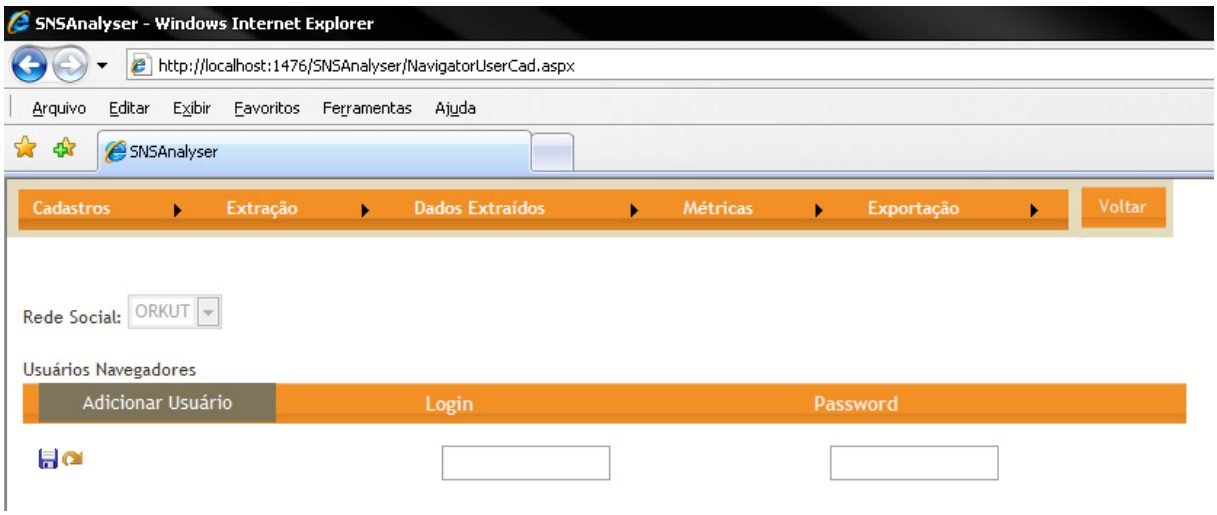


Figura 31 - Cadastro de Usuários Navegadores

A partir de agora, o SNSAnalyser está pronto para realizar a extração de redes sociais a partir de comunidades desse site de relacionamento. Entretanto, no exemplo escolhido, optou-se por incluir uma métrica customizada, para que também seja calculada para a rede extraída. A seção 5.3 relata isso.

5.3 CADASTRO DE MÉTRICA CUSTOMIZADA

A título de exemplo e também no intuito de trazer mais informações sobre a rede social, foi implementada uma métrica customizada, utilizando a *interface* IMetricFromResults. Essa métrica, aqui chamada de "Acumulated Degree", realiza a soma do

grau do nó com o grau dos nós vizinhos, atribuindo uma centralidade maior aos nós que estão conectados a outros também com grau alto. O Quadro 17 mostra o código utilizado para realizar esse cálculo.

```
using System;
using SNSAnalyser.Interfaces;
using System.Data;
using System.IO;
public class AcumulatedDegree : IMetricFromResults
{
    public string calculateFromResults(string networkData, string
distanceData, string resultsData)
    {
        DataSet dsNetwork = new DataSet();
        DataSet dsResults = new DataSet();
        DataSet dsAcumulatedDegree = new DataSet();
        dsNetwork.ReadXml(new StringReader(networkData));
        dsResults.ReadXml(new StringReader(resultsData));

        dsAcumulatedDegree.Tables.Add();
        dsAcumulatedDegree.Tables[0].Columns.Add("id",
typeof(System.Int32));
        dsAcumulatedDegree.Tables[0].Columns.Add("int_acumulatedDegree",
typeof(System.Int32));

        foreach (DataRow row in dsResults.Tables[0].Rows)
        {
            DataRow degreeRow = dsAcumulatedDegree.Tables[0].NewRow();
            degreeRow["id"] = Convert.ToInt32(row["id"]);
            int cdegree = Convert.ToInt32(row["degree"]);

            DataRow[] neighbors = dsNetwork.Tables[0].Select("id1 = " +
row["id"].ToString() + " OR id2 = " + row["id"].ToString());
            foreach (DataRow neighbor in neighbors)
            {
                DataRow neighborDegree;
                if (Convert.ToInt32(neighbor["id1"]) ==
Convert.ToInt32(degreeRow["id"]))
                {
                    neighborDegree = dsResults.Tables[0].Select("id = " +
neighbor["id2"].ToString())[0];
                }
                else
                {
                    neighborDegree = dsResults.Tables[0].Select("id = " +
neighbor["id1"].ToString())[0];
                }
                cdegree += Convert.ToInt32(neighborDegree["degree"]);
            }
            degreeRow["int_acumulatedDegree"] = cdegree;
            dsAcumulatedDegree.Tables[0].Rows.Add(degreeRow);
        }

        return dsAcumulatedDegree.GetXml();
    }
}
```

Quadro 17 - Exemplo de métrica customizada implementada

Pode-se observar no código exibido que o nome da coluna que contém os valores calculados para métrica implementada (“int_acumulatedDegree”) se inicia com o nome do tipo do dado seguido de *underline* e o nome que se quer dar à coluna da métrica. Apesar de ser opcional incluir o tipo do dado, esse é um recurso oferecido pelo SNSAnalyser para possibilitar que a exibição dos resultados das métricas customizadas também possa usufruir da funcionalidade de ordenação. Além de "int", que indica a utilização de uma coluna do tipo inteiro longo, também podem ser utilizados "dec" para decimal, "str" para string, "flt" para float e "dat" para data. A coluna “id”, também retornada pelo código, não precisa informar seu tipo para ser ordenada uma vez que a própria ferramenta já conhece essa informação.

A classe utilizada neste exemplo foi incluída num projeto chamado CustomMetric que, após a compilação, gerou a DLL CustomMetric.dll. Uma vez feito isso, essa métrica é incluída na ferramenta através do cadastro de métrica customizada, disponível no *menu* “cadastros > Métricas”. Nesta tela encontra-se o botão “Adicionar Métrica” que disponibiliza a interface para preenchimento dos dados da métrica que está sendo inserida no SNSAnalyser.

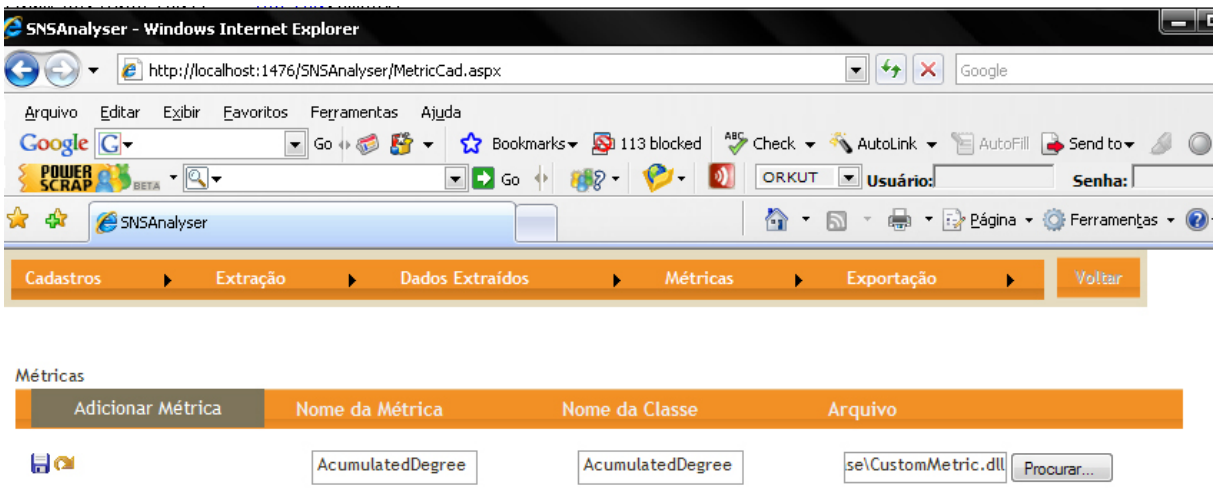


Figura 32 - Cadastro de Métricas Customizadas

Como pode ser visto na Figura 32, neste momento, é preenchido o nome da métrica e o nome completo da classe (incluindo seu *namespace*, caso possua), além de se informar a localização do código compilado (DLL) para que seja feito o *upload* do arquivo para o servidor da ferramenta.

5.4 EXTRAIR REDE SOCIAL

Antes de falar da extração propriamente dita, é importante explicar que, antes disso, duas configurações podem ser feitas na ferramenta, as quais irão influenciar diretamente a forma como o SNSAnalyser se conecta ao site de relacionamento.

A primeira delas é o número de *threads*. Configurável por meio da chave numberOfThreads colocada no web.config, o valor atribuído a ela define se a extração da rede social utilizará o modo Simple Thread ou Multi Thread. A essa chave deve ser atribuído um valor numérico que determinará o número de threads a serem criadas para realizar a extração. Caso não exista essa chave no web.config, ela seja inválida ou tenha o valor 1, o modo Simple Thread será usado, caso um número maior que 1 e menor ou iguala 100 seja utilizado, essa será a quantidade de *threads* disparada. Se for atribuído um número maior que 100 ao valor dessa chave, o SNSAnalyser utilizará, apesar disso, 100 *threads* a fim de evitar sobrecarga de conexões no servidor. O SNSAnalyser, entretanto, implementa uma pequena diferença entre o uso dessa chave com o valor 1 e a inexistência ou não atribuição de valor a ela. Quando a chave numberOfThreads não existe ou não possui valor, a ferramenta utiliza o modo Simple Thread aproveitando a requisição do usuário para realizar a extração e o usuário fica preso na tela até que a extração termine ou ele interrompa o processo. Já atribuindo o valor 1 à chave, o sistema se comporta da mesma forma como no modo Multi Thread, criando uma *thread* separada para realizar a extração e liberando a requisição do usuário para que ele possa continuar a utilizar o sistema enquanto o processo está sendo realizado.

A outra configuração que influencia no método de extração é a definição do uso de *proxy*. Para isso, pode-se colocar outra chave no web.config indicando o uso de Power Proxy, Power Manager ou nenhum deles. Desta forma, caso não exista a chave ProxyServer, ela não possua valor ou possua o valor “PL”, que indica uso de “*proxy* local”, o SNSAnalyser entenderá que não deve ser utilizado nem Power Proxy, nem Power Manager. Para utilizar, Power Proxy, deve-se informar nessa chave o IP ou os IPs dos servidores que serão utilizados como *proxy*, concatenados com o símbolo “:” e as portas que serão utilizadas para acessar esses servidores. Separando os IPs e as portas de cada servidor, utiliza-se o símbolo “|” e ao final de tudo, separado por “:”, deve ser incluída a identificação do modo de uso de Power Proxy que é “PP”. Assim, uma configuração de uso como Power Proxy de dois servidores com os 192.168.0.1 e 192.168.0.2, através das portas 8080 e 8081, respectivamente ficaria assim: “192.168.0.1:8080|192.168.0.2:8081:PP”

Já para a utilização do Power Manager, basta utilizar, ao invés dos IPs dos Power Proxies, o IP do Power Manager (podem ser os IPs dos Power Managers, caso exista mais de um Power Manager), seguido da indicação de uso do Power Manager, que é “PM”

Feitas as configurações, pode-se partir para a extração da rede social. O primeiro passo é definir a partir de quais comunidades se quer realizar a extração da rede social. No exemplo em questão, deseja-se realizar uma campanha para promoção de produtos ligados ao esporte esqui aquático para uma rede de lojas esportiva. Com esse objetivo, foi realizada uma busca no site de relacionamento por comunidades que tivessem alguma relação com esqui aquático e, portanto, utilizou-se o termo “esqui aquático” no filtro da pesquisa. Como resultado, foram retornadas quinze comunidades. No Quadro 18, podem ser vistos os identificadores, a quantidade de membros e os nomes dessas comunidades.

Id	Qtd. Membros	Nome
2999537	676	Ski Aquático e Wakeboard
472854	470	Ski aquatico Brasil
200800	194	Ski Aquático e Wakeboard
1340003	125	Ski Aquatico
9332237	122	Esqui aquatico é show.
6375886	107	Esqui Aquático
7837362	94	Quem ama Esqui aquatico
2935253	59	ESQUI AQUÁTICO A JATO
16124804	30	--ski aquatico em rifaina--
11817581	23	OS AMANTES DO SKI AQUATICO
35309897	19	esqui aquatico em PM
325931	15	Esqui aquático!
18794732	11	Esqui Aquático
3475763	4	Ski aquatico Portugal
40103432	2	esqui aquatico

Quadro 18 - Comunidades do Orkut retornadas na busca por Esqui Aquático

Com os resultados na mão, pode-se eliminar algumas comunidades em função da sua quantidade de membros ou de outras características específicas. Neste exemplo, as duas menores comunidades foram retiradas da amostra por possuírem apenas seis membros juntas (quatro de uma e dois da outra), além de que, a partir do nome de uma delas (Ski aquatico Portugal), pode-se supor que seus membros não são brasileiros, e, por isso, não fazem parte do público-alvo dessa campanha da rede de lojas esportivas localizada no Brasil.

Neste momento, é iniciada a extração. Para isso, o usuário acessa a interface de extração de membros a partir do *menu* “Extração > Membros”, seleciona a rede, e digita o número do identificador de uma das comunidades. Depois, repete-se esse processo até que os

membros das treze comunidades tenham sido extraídos. Ao final do processo, para o exemplo em questão, foram extraídos 1618 membros que farão parte da rede social que será construída. Nota-se que esse número não reflete o somatório dos totais de membros das redes extraídas, uma vez que existem membros que participam de mais de uma comunidade.

O passo seguinte é a extração de relacionamentos. Para iniciar o procedimento basta acessar o item do *menu* “Extração > Relacionamentos”. Não é necessária a exibição de nenhuma *interface*. O SNSAnalyser acessa o site de relacionamento, verifica as ligações existentes entre os membros extraídos das comunidades e monta a rede social que será analisada pela ferramenta.

A depender do número de *threads* utilizadas e da quantidade de membros extraídos esse processo pode demorar bastante, de forma que essa operação pode precisar ser interrompida pelo usuário. Para viabilizar essa ação, o SNSAnalyser tem um mecanismo de ajuste da base para que os membros que estavam sendo utilizados para a extração de relacionamentos retornem ao status de “waiting” e, com isso, voltem à fila de execução novamente.

Os status que um membro pode assumir são:

- waiting: caso ainda esteja na fila para extração dos relacionamentos.

- executing: caso a extração de relacionamentos esteja sendo executada no momento.

- ok: caso a extração de relacionamentos tenha sido realizada com sucesso.

- erro: caso tenha ocorrido erro na extração de relacionamentos.

Durante o procedimento de extração, pode-se querer visualizar o andamento do processo, em linhas gerais. Essa informação pode ser obtida acessando o Painel de Status através do *menu* “Dados Extraídos > Painel de Status”. Nesta tela, é exibido um sumário da quantidade de membros que se encontra em cada um dos status definidos. Além disso, duas funcionalidades estão disponíveis na interface do Painel de Status: retornar as extrações com erro para a fila e excluir registros com erro da rede de relacionamento. A Figura 33 mostra o Painel de Status da extração desse exemplo em um determinado momento.

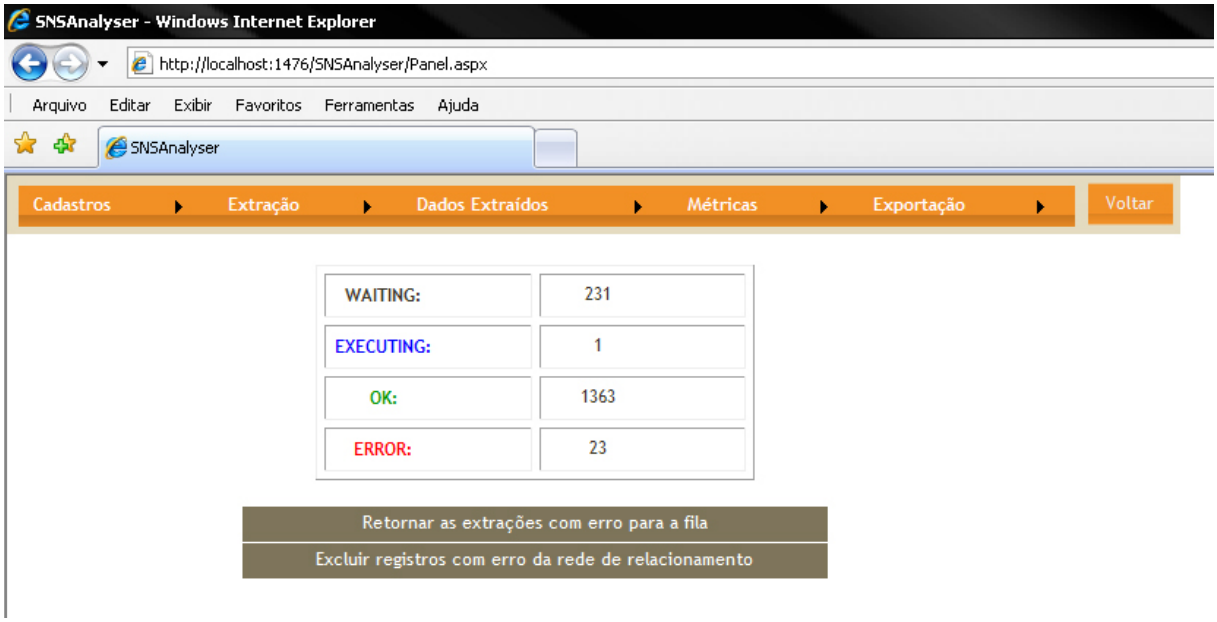


Figura 33 - Interface de exibição dos membros e relacionamentos extraídos

Ao final do procedimento de extração da rede social, no exemplo em questão, havia 442 itens com erro, pelo fato do Orkut™ ter ficado indisponível por um período de 20 minutos. Nesse contexto, foi acionada a opção de retornar as extrações com erro para a fila e solicitada a extração de relacionamentos novamente. Ao final do procedimento, restaram 263 usuários com status de erro. Avaliando, esses usuários, pôde-se perceber que o Orkut™ disponibiliza uma opção para usuários de determinados países onde ele pode optar para não exibir quem são os usuários com quem ele se relaciona. Desta forma, ao tentar obter os relacionamentos desses usuários, o SNSAnalyser recebe uma tela informando: "As configurações de privacidade do usuário impedem a visualização do conteúdo nesta página". Em função disso, optou-se por excluir esses usuários da rede de relacionamento construída.

O SNSAnalyser também permite que a qualquer momento se possa visualizar os membros e relacionamentos extraídos de forma detalhada através das opções "Exibir Membros", "Exibir Relacionamentos" e "Exibir Membros e Relacionamentos" do *menu* "Dados Extraídos". Como a interface "Exibir Membros e Relacionamentos" é uma junção das interfaces "Exibir Membros" e "Exibir Relacionamentos", apenas ela será exibida neste documento. Na Figura 34, pode-se ver essa tela apresentando os membros extraídos no primeiro grid, inclusive relatando o status da extração de cada membro, e os relacionamentos no segundo grid.

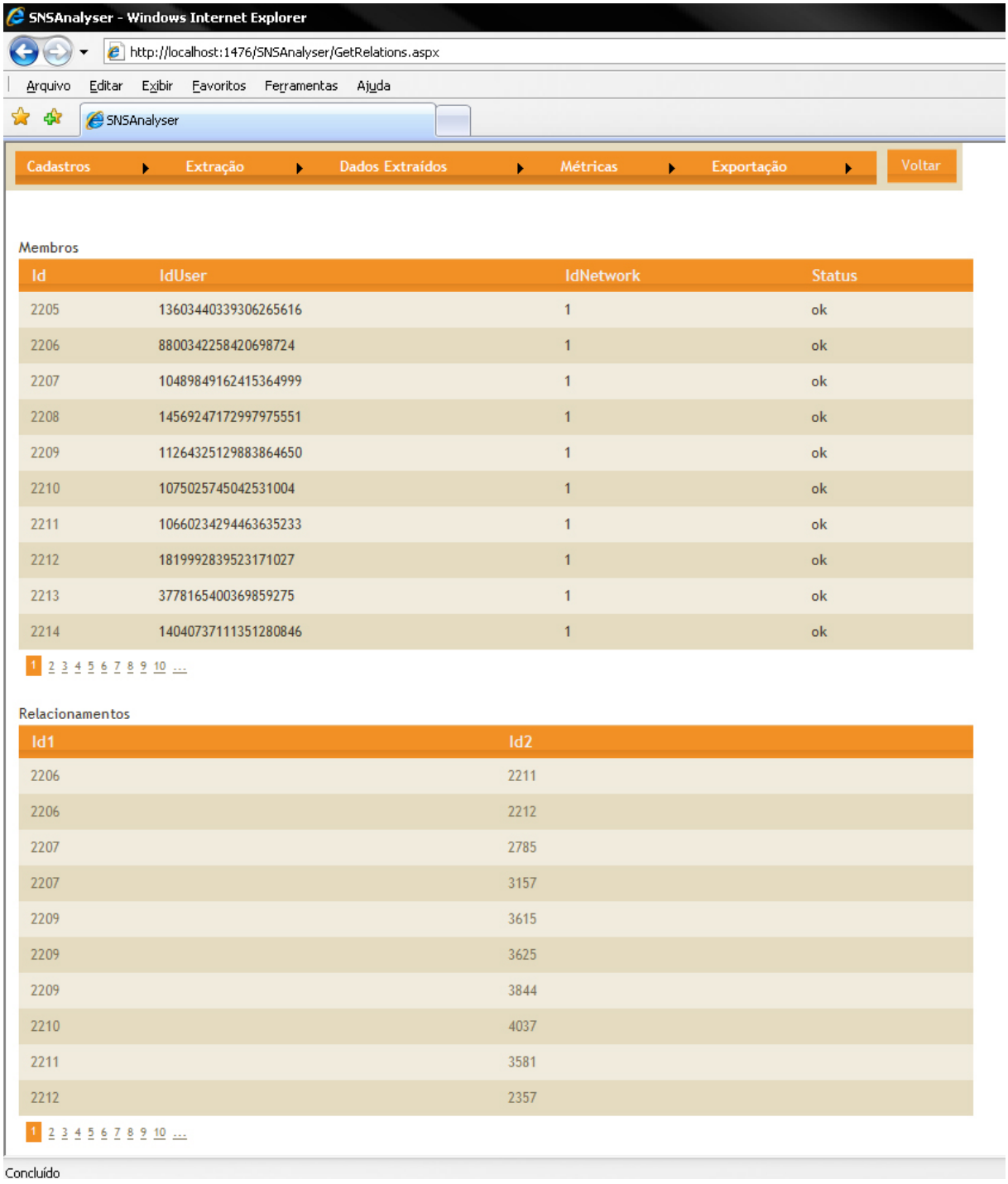
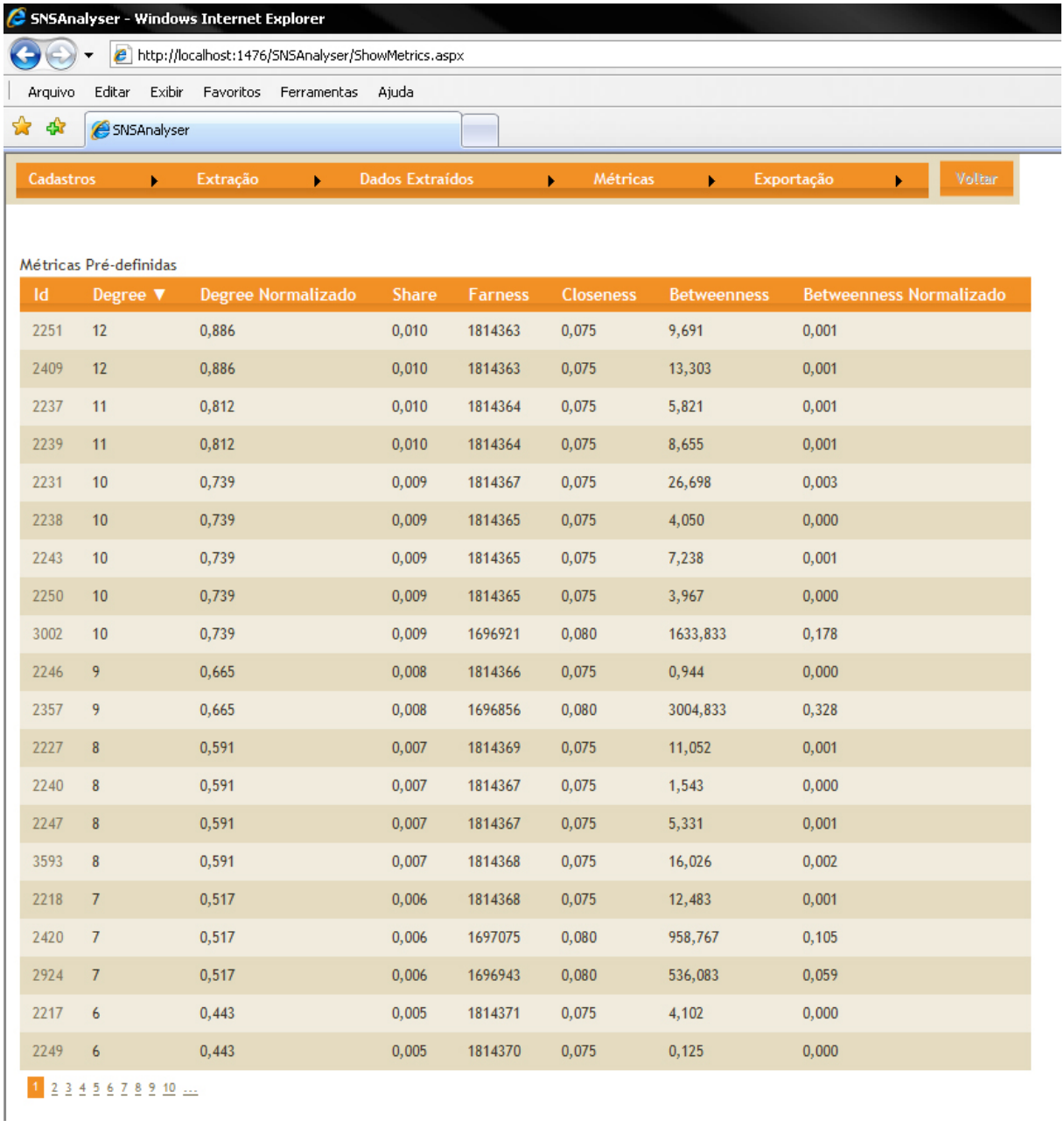


Figura 34 - Interface de exibição dos membros e relacionamentos extraídos

Uma vez finalizada a extração, parte-se para o cálculo de métricas que será relatado na seção seguinte.

5.5 CÁLCULO DE MÉTRICAS

O cálculo das métricas pré-definidas está acessível através do *menu* “Métricas > Pré-definidas > Calcular”. Já as métricas customizadas podem ser acessadas através do *menu* “Métricas > Customizadas > Calcular”. Ao final de cada cálculo, os resultados são exibidos. Entretanto, uma vez calculado, pode-se verificar seus resultados a qualquer momento através dos *menus* “Métricas > Pré-definidas > Exibir” e “Métricas > Customizadas > Exibir” para o resultado dos cálculos de métricas pré-definidas e customizadas respectivamente.



Id	Degree ▼	Degree Normalizado	Share	Farness	Closeness	Betweenness	Betweenness Normalizado
2251	12	0,886	0,010	1814363	0,075	9,691	0,001
2409	12	0,886	0,010	1814363	0,075	13,303	0,001
2237	11	0,812	0,010	1814364	0,075	5,821	0,001
2239	11	0,812	0,010	1814364	0,075	8,655	0,001
2231	10	0,739	0,009	1814367	0,075	26,698	0,003
2238	10	0,739	0,009	1814365	0,075	4,050	0,000
2243	10	0,739	0,009	1814365	0,075	7,238	0,001
2250	10	0,739	0,009	1814365	0,075	3,967	0,000
3002	10	0,739	0,009	1696921	0,080	1633,833	0,178
2246	9	0,665	0,008	1814366	0,075	0,944	0,000
2357	9	0,665	0,008	1696856	0,080	3004,833	0,328
2227	8	0,591	0,007	1814369	0,075	11,052	0,001
2240	8	0,591	0,007	1814367	0,075	1,543	0,000
2247	8	0,591	0,007	1814367	0,075	5,331	0,001
3593	8	0,591	0,007	1814368	0,075	16,026	0,002
2218	7	0,517	0,006	1814368	0,075	12,483	0,001
2420	7	0,517	0,006	1697075	0,080	958,767	0,105
2924	7	0,517	0,006	1696943	0,080	536,083	0,059
2217	6	0,443	0,005	1814371	0,075	4,102	0,000
2249	6	0,443	0,005	1814370	0,075	0,125	0,000

Figura 35 - Exibição de métricas pré-definidas ordenada pelo grau

Como pode ser vista na Figura 35, a relação de métricas pré-definidas está ordenada pelo grau. Isso porque na interface de exibição dos resultados, o usuário pode ordenar a tabela por todas as suas colunas para facilitar o reconhecimento de usuários mais centrais. Também é permitido visualizar informações do membro da rede social, bastando para isso clicar na primeira coluna que apresenta o id do membro em forma de link. Essas observações valem tanto para os resultados das métricas pré-definidas, quanto das métricas customizadas. A Figura 36 mostra a interface de métricas customizadas com o detalhamento do usuário mais interessante, pois possui o maior valor, segunda a ótica dessa medida.

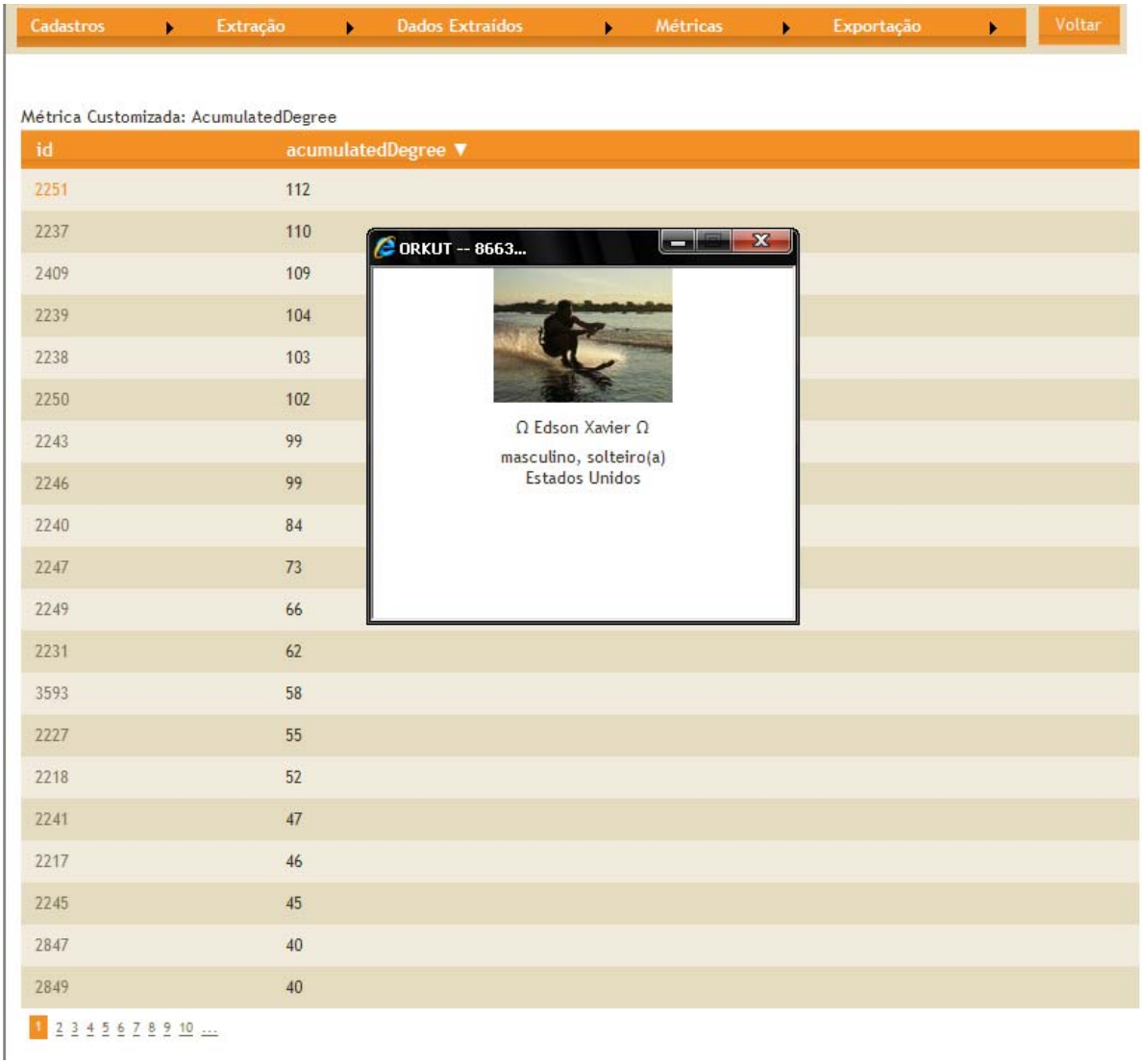


Figura 36 - Exibição de resultados de métricas customizadas e dos atributos de um usuário

Com os resultados dos cálculos em mãos, o pesquisador pode fazer a análise e avaliar que usuários são mais interessantes para ele trabalhar. Neste exemplo, optou-se por fazer uma

seleção dos 20 melhores usuários de acordo com cada métrica, dos quais 10 serão escolhidos para fazer um *test drive* nos novos aparelhos de esqui aquático que a empresa está promovendo. Os demais receberão divulgação eletrônica apenas.

Neste contexto, os quinze melhores resultados de cada medida foram colocados numa tabela e, após análise, dez usuários foram destacados, como pode ser visto no quadro 19.

Id User	Degree	Id User	Degree Normaliz.	Id User	Between.	Id User	Between. Normaliz.	Id User	Share	Id User	Farness	Id User	Closen.	Id User	Acum. Degree
2251	**12**	**2251**	**0,886**	**2357**	**3004,833**	**2357**	**0,328**	**2237**	**0,01**	**2357**	**1696856**	2206	0,08	**2251**	**112**
2409	**12**	**2409**	**0,886**	**2861**	**2260**	**2861**	**0,247**	**2239**	**0,01**	**2861**	**1696871**	2211	0,08	**2237**	**110**
2237	**11**	**2237**	**0,812**	**3002**	**1633,833**	**3002**	**0,178**	**2251**	**0,01**	**3259**	**1696886**	2212	0,08	**2409**	**109**
2239	**11**	**2239**	**0,812**	**3259**	**1383**	**3259**	**0,151**	**2409**	**0,01**	3318	1696910	2258	0,08	**2239**	**104**
2231	**10**	**2231**	**0,739**	**2306**	**1362**	**2306**	**0,149**	**2231**	**0,009**	2311	1696915	2303	0,08	2238	103
2238	10	2238	0,739	2311	1235,667	2311	0,135	2238	0,009	**3002**	**1696921**	2305	0,08	2250	102
2243	10	2243	0,739	3530	1200,5	3530	0,131	2243	0,009	3131	1696921	2306	0,08	2243	99
2250	10	2250	0,739	3031	1180,5	3031	0,129	2250	0,009	3630	1696921	2307	0,08	2246	99
3002	**10**	**3002**	**0,739**	2651	1011	2651	0,11	**3002**	**0,009**	3357	1696925	2311	0,08	2240	84
2246	9	2246	0,665	2420	958,767	2420	0,105	2246	0,008	2924	1696943	2312	0,08	2247	73
2357	**9**	**2357**	**0,665**	3630	876,167	3630	0,096	**2357**	**0,008**	3766	1696947	2313	0,08	2249	66
2227	8	2227	0,591	2258	857	2258	0,094	2227	0,007	2651	1696950	2314	0,08	**2231**	**62**
2240	8	2240	0,591	2317	842	2317	0,092	2240	0,007	2212	1696951	2315	0,08	3593	58
2247	8	2247	0,591	2321	821,5	2321	0,09	2247	0,007	2306	1696957	2317	0,08	2227	55
3593	8	3593	0,591	2314	772,567	2314	0,084	3593	0,007	3031	1696971	2320	0,08	2218	52

Quadro 19 - Quinze melhores resultados de cada medida com os dez usuários selecionados destacados com cor

Os usuários com identificadores 2251, 2409, 2237, 2239 e 2231 são os cinco usuários com maior grau de centralidade (*degree*), incluindo o grau de centralidade normalizado e a participação relativa (*share*) e, por isso, foram selecionados. Os outros cinco usuários selecionados são os que possuem identificador 2357, 2861, 3002, 3259 e 2306. Eles foram os que apresentaram maior grau de intermediação (*betweenness*) e grau de intermediação normalizado. Dentre eles, três possuem também o menor grau de distanciamento (*farness*) e quatro tiveram os maiores resultados no cálculo da métrica customizada *Acumulated Degree*. O grau de proximidade apresentou números aproximadamente iguais e por isso não foi utilizado como critério na seleção de usuários.

Apesar dos dados terem sido suficientes para a análise e a seleção dos usuários já estar feita, existe a opção de se querer avaliar outras medidas que não são oferecidas pelo SNSAnalyser nem foram implementadas para ela como uma métrica customizada. Neste caso, faz-se necessária a realização da exportação dos dados. Em caráter ilustrativo, a seção 5.6 aborda esse assunto.

5.6 EXPORTAÇÃO DE DADOS

A versão atual do SNSAnalyser possibilita 3 tipos de exportação. A primeira opção é para o UCINet (BORGATTI; EVERETT; FREEMAN, 2002), uma das principais ferramentas disponíveis para análise de redes sociais, conforme foi visto no capítulo 2. As outras duas são para Excel™, planilha eletrônica da Microsoft™, com a diferença que uma é mais simples e a outra mais completa.

Para realizar a exportação, basta selecionar uma das três opções do *menu* "Exportação", conforme é indicado na Figura 37.

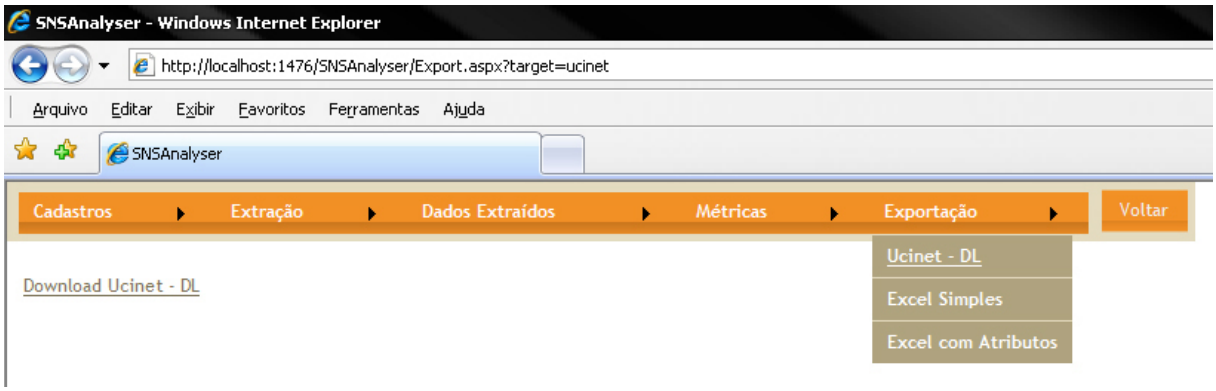


Figura 37 - Menu de exportação e link para download do arquivo gerado

Como pode ser visto na figura acima, após selecionar uma das opções do *menu*, um link para *download* do arquivo gerado. O formato e o conteúdo desses arquivos já foram abordados na seção 4.3.8 do presente documento.

5.7 CONSIDERAÇÕES FINAIS

Nesse capítulo foi apresentado um exemplo de extração e análise de redes sociais a partir de comunidades existentes em sites de relacionamento, utilizando, para isso, a ferramenta SNSAnalyser. Neste processo, utilizou-se o Orkut™, site de relacionamento da empresa Google™, como base para recuperação dos usuários e seus relacionamentos.

No intuito de oferecer uma visão mais prática do uso da ferramenta e dos conhecimentos que dela provém, simulou-se a existência de uma empresa de venda de materiais esportivos que tem interesse em realizar a promoção de um novo produto para apreciadores do esqui aquático. Desta forma, as comunidades do Orkut™ relacionadas a isso foram o alvo das extrações e formam o cerne da rede social criada.

As atividades executadas na utilização da ferramenta foram detalhadas neste capítulo, desde o cadastramento do site de relacionamento até a análise das métricas da rede social montada, convergindo para a seleção dos usuários que formam o público-alvo mais interessante para a promoção da empresa de materiais esportivos.

Os resultados obtidos evidenciaram que o SNSAnalyser atende ao objetivo para qual foi implementado, indicando que a solução proposta pela presente pesquisa – extrair e analisar redes sociais a partir de comunidades existentes em sites de relacionamento – apresenta-se como passível de aplicabilidade na área de análise de redes sociais. A ferramenta apresentou resultados que podem auxiliar empresas e profissionais a selecionar pessoas dentro de um determinado perfil que se destacam dentro da rede, apresentando muitas aplicações científicas e comerciais.

No próximo capítulo será apresentada a conclusão deste trabalho, incluindo considerações sobre contribuições da pesquisa e perspectivas.

6 CONCLUSÃO

A noção de redes sociais e os métodos de análise dessas redes têm sido bastante usados na comunidade científica para analisar relacionamentos entre entidades sociais e os padrões e implicações desses relacionamentos (WASSERMAN; FAUST, 1994), apresentando-se como um instrumento de análise de dados distinto dos métodos estatísticos tradicionais.

A análise de redes sociais foi incorporada na ciência social como forma de subsidiar pesquisadores na descrição de fenômenos empíricos onde se dá importância às interações entre os atores de um determinado contexto social. A análise de redes sociais não toma como unidade de análise o ator individual que faz parte da rede em estudo, mas a coleção de atores ou indivíduos e as suas interações.

Segundo Marteleto (2001, p. 73), as redes nas ciências sociais designam normalmente os movimentos fracamente institucionalizados, reunindo indivíduos e grupos em uma associação cujos termos são variáveis e sujeitos a uma re-interpretação em função dos limites que pesam sobre suas ações.

Aproveitando o crescimento e a solidificação do uso de sites de relacionamento, é possível tirar proveito de uma ótima oportunidade para a realização de estudos em redes sociais compostas por membros de comunidades existentes nesses sistemas. Usuários dos sites de relacionamentos e seus relacionamentos com outros participantes de uma mesma comunidade podem ser extraídos, formando uma rede social com pessoas que apresentam características ou interesses em comum.

Para aproveitar essa oportunidade, é necessária uma ferramenta que permita a extração e análise de redes sociais a partir de comunidades existentes em sites de relacionamento. Assim, a possibilidade de explorar e analisar novas redes compostas por usuários selecionados por um perfil ou característica específica apresenta-se como uma opção real, o que viabiliza a seleção de pessoas, seja para promoções de produto ou quaisquer outros fins, de uma forma antes inexplorada.

O presente trabalho apresentou uma proposta para o trato desta questão. Trata-se de uma solução para extração de informações de quaisquer sites de relacionamento, além da

análise da rede social montada. Para tanto, foram utilizadas as estratégias para extração de redes sociais de grandes sites de relacionamento, abordadas no capítulo 3.

Visando a análise da aplicabilidade da solução proposta, foi desenvolvida uma ferramenta descrita no capítulo 4: o SNSAnalyser. Nela, além da extração de redes sociais a partir de comunidades existentes em sites de relacionamento, foram incluídas métricas pré-definidas, selecionadas conforme critérios estabelecidos no capítulo 2, além de possibilitadas exportação dos dados extraídos e inclusão de métricas customizadas. Um exemplo de uso da ferramenta, em caráter prático e demonstrativo foi detalhado no capítulo 5, tomando como base o site de relacionamento Orkut™. Nesse contexto, verificou-se o atendimento da ferramenta ao objetivo para o qual foi desenvolvida, bem como possíveis sugestões para melhorias futuras.

6.1 CONTRIBUIÇÕES

A principal contribuição deste trabalho consiste em propor uma solução para extração e análise de redes sociais a partir de comunidades existentes em sites de relacionamento.

Durante o desenvolvimento desse trabalho não foram encontradas ferramentas que realizem extração de comunidades existentes em sites de relacionamento, o que torna a contribuição deste trabalho ainda maior.

Com relação à análise de métricas, não foi intenção deste trabalho concorrer com as ferramentas existentes e já estabelecidas no mercado e, por isso, o SNSAnalyser disponibiliza um número inferior de métricas em relação a outras ferramentas como UCINet e NetMiner. Entretanto, a possibilidade de inclusção de métricas customizadas implementadas pelo usuário se apresenta como um diferencial da ferramenta.

Outras contribuições deste trabalho podem ser destacadas:

- Estudo comparativo entre os métodos de extração de redes sociais
- Discussão de estratégias para extrair redes sociais a partir de sites de relacionamento.
- Definição e implementação de uma ferramenta apta a implantar a solução supra mencionada.

- Criação de interfaces de comunicação da ferramenta com o mundo exterior, permitindo a inclusão de novas métricas customizadas no sistema ou a exportação dos dados extraídos para análise em outras ferramentas.
- Pesquisa e comparação de ferramentas para análise de redes sociais.
- Apresentação da infra-estrutura de *proxy*, *manager* e *spider*, utilizadas no SNSAnalyser, que facilitam a customização da ferramenta para extrações em quaisquer sites de relacionamento e auxiliam o desvio de bloqueios durante a recuperação das informações.

6.2 LIMITAÇÕES DO TRABALHO DESENVOLVIDO

Durante o desenvolvimento do trabalho algumas dificuldades surgiram e delimitaram limites que a solução proposta respeita.

A principal limitação do trabalho diz respeito às restrições impostas pelos sites de relacionamentos no uso de suas interfaces e nas regras de privacidade. A tela de exibição de membros de comunidades do Orkut, por exemplo, exibe no máximo mil membros, o que impede que sejam extraídos os demais membros de uma comunidade com um número de integrantes maior que esse limite. Essa é uma restrição da interface do Orkut para todos os seus usuários e o SNSAnalyser, por atuar como um usuário do site de relacionamento, acessando as páginas através do *spider*, é obrigado a respeitar essa regra.

Atualmente, nos principais sites de relacionamentos, é possível restringir a visualização das fotos e recados do usuário para que apenas as pessoas relacionadas a ele consigam acessar essas informações. Caso esse tipo de regra de privacidade fosse implementado pelos sites de relacionamentos, a ferramenta se limitaria à extração dos relacionamentos dos usuários que não adotassem essa regra.

6.3 PERSPECTIVAS

A elaboração do exemplo de uso da ferramenta não apenas ratificou a aplicabilidade do SNSAnalyser como ferramenta de extração e análise de redes sociais a partir de comunidades existentes em sites de relacionamento como também indicou melhorias passíveis de implementação em trabalhos futuros.

A principal extensão sugerida consiste na implementação de uma novo modelo de visualização dos resultados. O formato de tabelas, associado à utilização de paginação, satisfaz quando a quantidade de usuários relevantes é pequena, mas dificulta a analisa para volumes maiores.

Assim, apresenta-se como candidata natural a futuros trabalhos a implementação de uma nova forma de visualização dos resultados, incluindo uma abordagem visual mais arrojada, com a utilização de grafos, nós e arestas, além de filtros e agrupamentos por rede ou atributo do usuário para possibilitar outras análises ou simplesmente aumentar ou diminuir o volume de registros exibidos de acordo com os resultados das medições.

Além dessa sugestão, podem ser listados os seguintes trabalhos futuros relacionados à evolução da solução proposta:

- Incluir novas métricas pré-definidas para auxiliar a análise da rede social.
- Criar funcionalidade que permita contato do usuário do SNSAnalyser com os integrantes de seu interesse na rede, possibilitando, inclusive, a promoção dos produtos através de mensagens para esses usuários relevantes no site de relacionamento.
- Incluir outros formatos de exportação de dados extraídos e implementar a exportação dos resultados das medições.
- Pesquisar formas de atribuir pesos e utilizá-los para usuários pertencentes a mais de uma comunidade extraída.

6.4 CONSIDERAÇÕES FINAIS

Este trabalho apresentou uma solução para extração e análise de redes sociais a partir de comunidades existentes em sites de relacionamento. Durante o trabalho foram apresentadas a fundamentação teórica e revisão da literatura relacionadas ao tema, a descrição detalhada do trabalho desenvolvido, o exemplo de uso da ferramenta SNSAnalyser, as contribuições obtidas e também as perspectivas para evolução da solução proposta.

REFERÊNCIAS

ADAMIC, L. A.; ADAR, E. Friends and neighbors on the web. **Social Networks**, 2003, v. 25, n. 3, p. 211-230.

AGNA. **Agna homepage.** Disponível em: <http://www.geocities.com/imbenta/agna/index.htm>. Acesso em: 12 jun. 2008.

ALEXA. **The web information company.** Disponível em: <http://www.alexa.com>. Acesso em: 11 mar. 2008.

BENTA, M. I. **Agna 2.1 user manual.** 1st ed. [S.l.: s.n.], 2003. Disponível em: <http://www.geocities.com/imbenta/agna/doc/User_Manual.htm>. Acesso em: 27 jun. 2008.

BORGATTI, S. P.; EVERETT, M. G.; FREEMAN, L.C. **UCINet 6 for Windows:** software for social network analysis. Harvard, MA: Analytic Technologies, 2002. Disponível em: <http://www.analytictech.com>. Acesso em: 22 jan. 2008.

BUSCH, P. A.; RICHARDS, D.; DAMPNEY, C. N. G. Visual mapping of articulable tacit knowledge. In: AUSTRALIAN SYMPOSIUM ON INFORMATION VISUALISATION, 2001, Sydney, Australia. **Proceedings…** Sydney: Australian Computer Society (ACS), 2001, p.37-47. Disponível em: <http://crpit.com/confpapers/CRPITV9Busch.pdf>. Acesso em: 1 jul. 2008.

COHEN, S. et al. Social integration and health: the case of the common cold. **Journal of Social Structure**, 2000. v.1, n. 3, p. 1-7. Disponível em: <http://www.cmu.edu/joss/content/articles/volume1/cohen.html>. Acesso em: 1 jul. 2008.

COSTA, L. F. et al. Characterization of complex networks: a survey of measurements. **Advances in Physics**, 2007, v. 56, n. 1, p. 167-242.

CROSS, R.; BORGATTI, S. P.; PARKER, A. Beyond answers: dimensions of the advice network. **Social Networks**, 2001, v. 23, n. 3, p. 215-235.

CROSS, R.; PARKER, A. **The hidden power of social networks:** understanding how work really gets done in organizations. 1st ed. Boston, Massachusetts: Harvard Business School Press, 2004.

CULOTTA, A.; BEKKERMAN, R.; MCCALLUM, A. Extracting social networks and contact information from email and the web. In: CONFERENCE ON EMAIL AND ANTI-SPAM, 1., 2004, Mountain View, California, USA. **Proceedings…** Mountain View: CEAS. Disponível em: <http://www.ceas.cc/papers-2004/176.pdf>. Acesso em: 30 jul. 2008.

DILIGENTI, M. et al. Focused crawling using context graphs. In: INTERNATIONAL CONFERENCE ON VERY LARGE DATABASES, 26., 2000, Cairo, Egypt. **Proceedings…** San Francisco: Morgan Kaufmann Publishers, 2000, p. 527-534.

ENGLISH, J. **Ada 95:** the craft of object-oriented programming. Prentice Hall, 1997. Disponível em: <http://www.cmis.brighton.ac.uk/~je/adacraft/glossary.htm>. Acesso em: 4 abr. 2008.

FACEBOOK. **Facebook.** Disponível em: <http://www.facebook.com>. Acesso em: 1 ago. 2008

FININ, T. et al. Social networking on the semantic web. **The Learning Organization Journal**, 2005, v. 12, nº 5, p. 418-435.

FOWLER, M. **Patterns of enterprise application architecture.** 1st ed. Boston: Massachusetts: Addison-Wesley Professional, 2002.

FREEMAN, L. C. Some antecedents of social network analysis. **Connections**, 1996, v. 19, n. 1, p. 39-42.

FREEMAN, L. C., Visualizing social networks. **Journal of Social Structure**, 2000, v. 1, n. 1, p. 1-15. Disponível em: <http://www.cmu.edu/joss/content/articles/volume1/Freeman.html>. Acesso em: 30 jun. 2008.

G1. **O portal de notícias da Globo.** Disponível em: <http://g1.globo.com/>. Acesso em: 31 jan. 2008.

GAMMA, E. et al. **Padrões de projeto:** soluções reutilizáveis de software orientado a objetos. 1 ed. Porto Alegre: Bookman, 2000.

GOOGLE. **Google.** Disponível em: <http://www.google.com>. Acesso em: 28 fev. 2008.

GRUDIN, J. Groupware and social dynamics: eight challenges for developers. **Communications of the ACM**, 1994, v. 37, n. 1, p. 92-105.

HAMASAKI, M. et al. Community focused social network extraction. In: ASWC ASIAN SEMANTIC WEB CONFERENCE, 2006, Beijing, China. **Proceedings…** Berlin: Springer, 2006, p. 155-161.

HANNEMAN, R. **Introduction to social network methods.** 1st ed. Riverside, CA: University of California, 2001. Disponível em: <http://faculty.ucr.edu/~hanneman/>. Acesso em: 12 abr. 2008.

HARADA, M.; SATO, S.; KAZAMA, K. Finding authoritative people from the web. In: JCDL JOINT CONFERENCE ON DIGITAL LIBRARIES, 2004, Tuscon, AZ, USA. **Proceedings…** New York: ACM, 2004, p. 306-313.

HEINONEN, O.; HATONEN, K.; KLEMETTINEN, K. WWW robots and search engines. In: SEMINAR ON MOBILE CODE, 1996, Otaniemi, Espoo, Finland. **Report TKO-C79.** Otaniemi: Helsinki University of Technology, 1996.

HOPE, T.; NISHIMURA, T.; TAKEDA, H. An integrated method for social network extraction. In: INTERNATIONAL CONFERENCE ON WORLD WIDE WEB, 15., 2006, Edinburgh, Scotland. **Proceedings…** New York: ACM Press, 2006, p. 845-846.

HUFFAKER, B.; NEMETH, E.; CLAFFY, K. Otter: a general purpose network visualization tool. In: INET ANNUAL CONFERENCE OF THE INTERNET SOCIETY, 9., 1999, San Jose, California, USA. **Proceedings…** Disponível em: <http://www.isoc.org/inet99/proceedings/4h/4h_3.htm>. Acesso em: 1 jul. 2008.

FREEMAN, L. C. Some antecedents of social network analysis. **Connections**, 1996, v. 19, n. 1, p. 39-42.

HUISMAN, M.; VAN DUIJN, M.A.J. StOCNET: **s**oftware for the statistical analysis of social networks. **Connections**, 2003, v. 25, n. 1, p. 7-26.

IICM. **Institute for information systems and computer media.** Disponível em: <http://www.iicm.tugraz.at/>. Acesso em: 7 abr. 2008.

KADUSHIN, C. **Introduction to social network theory**. 1st ed. [S.l.: s.n.], 2004.

KAUTZ, H.; SELMAN, B.; SHAH, M. The hidden web. **AI Magazine**, 1997, v. 18, n. 2, p. 27-36.

KOSTER, M. Robots in the web: threat or treat? **ConneXions**, 1995, v. 9, n. 4, p. 2-12.

KRACKPLOT. **Krackplot software.** Disponível em: <http://www.andrew.cmu.edu/user/krack/krackplot.shtml>. Acesso em: 30 jul. 2008.

MARTELETO, R. M. Análise de redes sociais: aplicação nos estudos de transferência de informação. **Ciência da Informação**, Brasília, 2001, v. 30, n. 1, p. 71-81.

MATSUO, Y. et al. Polyphonet: an advanced social network extraction system. In: INTERNATIONAL CONFERENCE ON WORLD WIDE WEB, 15., 2006, Edinburgh, Scotland. **Proceedings…** New York: ACM Press, 2006, p. 397-406.

MCDONALD, D. W. Recommending collaboration with social networks: a comparative evaluation. In: ACM CONFERENCE ON HUMAN FACTORS IN COMPUTING SYSTEMS, 2003, Ft. Lauderdale, Florida. **Proceedings…** New York: ACM Press, 2003, p. 593-600.

MICROSOFT. **Microsoft Corporation**. Disponível em: <http://www.microsoft.com>. Acesso em: 19 nov. 2007.

MIKA, P. Flink: semantic web technology for the extraction and analysis of social networks. **Journal of Web Semantics**, 2005, v. 3, n. 2, p. 211-223.

MOLINA, J. L. El estudio de las redes personales: contribuciones, métodos y perspectives. **Empiria**, 2005, n. 10, p. 71-105.

MOODY, J.; MCFARLAND, D. A.; BENDER-DEMOLL, S. Visualizing network dynamics. **American Journal of Sociology**, 2005, v. 110, n. 4, p. 1206-1241.

MORTON, S. C. et al. Managing the informal organization: conceptual model. **International Journal of Productivity and Performance Management**, 2004, v. 53, n. 3, p. 214-232.

MSDN. **MSDN home page.** Disponível em: <http://msdn.microsoft.com>. Acesso em: 11 mar. 2008.

MYSPACE. **MySpace.** Disponível em: <http://www.myspace.com/>. Acesso em: 15 ago. 2008.

NETMINER. **Social network analysis software**. Disponível em: <http://www.netminer.com>. Acesso em: 15 ago. 2008.

NETVIS. **Dynamic visualization of social networks.** Disponível em: <http://www.netvis.org/>. Acesso em: 12 jun. 2008.

OMG. **UML resource page.** Disponível em: <http://www.uml.org/>. Acesso em: 15 ago. 2008.

ORKUT. **Orkut.** Disponível em <http://www.orkut.com>. Acesso em: 27 fev. 2008.

OTTE, E.; ROUSSEAU, R. Social network analysis: a powerful strategy, also for information sciences. **Journal of Information Science**, Thousand Oaks, 2002, v. 28, n. 6, p. 441-453.

PAJEK. **Networks / Pajek.** Disponível em: <http://vlado.fmf.uni-lj.si/pub/networks/pajek/>. Acesso em: 16 ago. 2008.

PARKER, A.; CROSS, R.; WALSH, D. Improving collaboration with social network analysis: leveraging knowledge in the informal organization. **Knowledge Management Review**, 2001, v. 4, n. 2, p. 24-30.

PENTLAND, A. Socially aware computation and communication. **Computer**, 2005, vol. 38, nº 3, p. 33-40.

PERER, A. Making sense of social networks. In: CONFERENCE ON HUMAN FACTORS IN COMPUTING SYSTEMS, 2006, Montréal, Québec, Canada. **Proceedings…** New York: ACM Press, 2006, p. 1779-1782.

POSTGRESQL. **PostgreeSQL:** the world's most advanced open source database. Disponível em: <http://www.postgresql.org/>. Acesso em: 23 mar. 2008.

POWER.COM. **Power.com.** Disponível em: <http://www.power.com>. Acesso em: 14 ago. 2008.

SCIENCE. **Science Plus Group:** software for social science, psychology and more. Disponível em: <http://www.scienceplus.nl/>. Acesso em: 3 ago. 2008.

STOCNET. **StOCNET.** Disponível em: <http://stat.gamma.rug.nl/stocnet/>. Acesso em: 1 jul. 2008.

TANENBAUM, A. S. **Sistemas operacionais modernos.** 2 ed. São Paulo: Prentice Hall, 2003.

TOUCHGRAPH. **TouchGraph home.** Disponível em <http://www.touchgraph.com/>. Acesso em: 9 jun. 2008.

TYNA. **tYNA control panel.** Disponível em: <http://tyna.gersteinlab.org>. Acesso em: 9 jun. 2008.

W3C. **World Wide Web Consortium.** Disponível em: <http://www.w3.org/>. Acesso em: 3 ago. 2008.

WASSERMAN, S.; FAUST, K. **Social network analysis:** methods and applications. 1st ed. Cambridge, New York: Cambridge University Press, 1994.

WIKIPEDIA. **Wikipédia:** a enciclopédia livre. Disponível em: <http://pt.wikipedia.org>. Acesso em: 22 nov. 2007.

YED. **Java Graph Editor.** Disponível em: <http://www.yworks.com/en/products_yed_about.html>. Acesso em: 2 jul. 2008.

YIP, K. et al. The tYNA platform for comparative interactomics: a web tool for managing, comparing and mining multiple networks. **Bioinformatics**, 2006, v. 22, n. 23, p. 2968-2970.

APÊNDICE A – ARTEFATOS DA FERRAMENTA SNSANALYSER

Abaixo, seguem artefatos elaborados durante o processo de desenvolvimento da ferramenta SNSAnalyser, sendo alguns destes modelos elaborados segundo a notação UML™ (*Unified Modeling Language*).

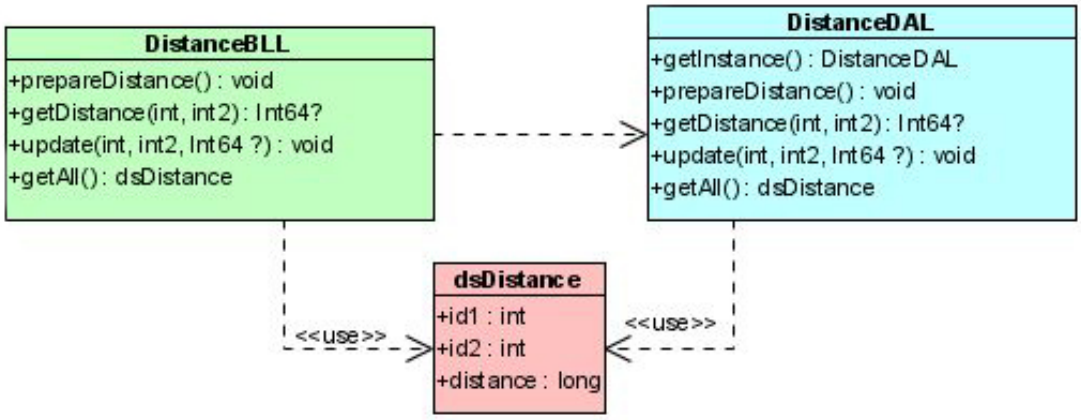


Figura 38 - Diagrama de Classes – DistanceBLL, DistanceDAL e dsDistance

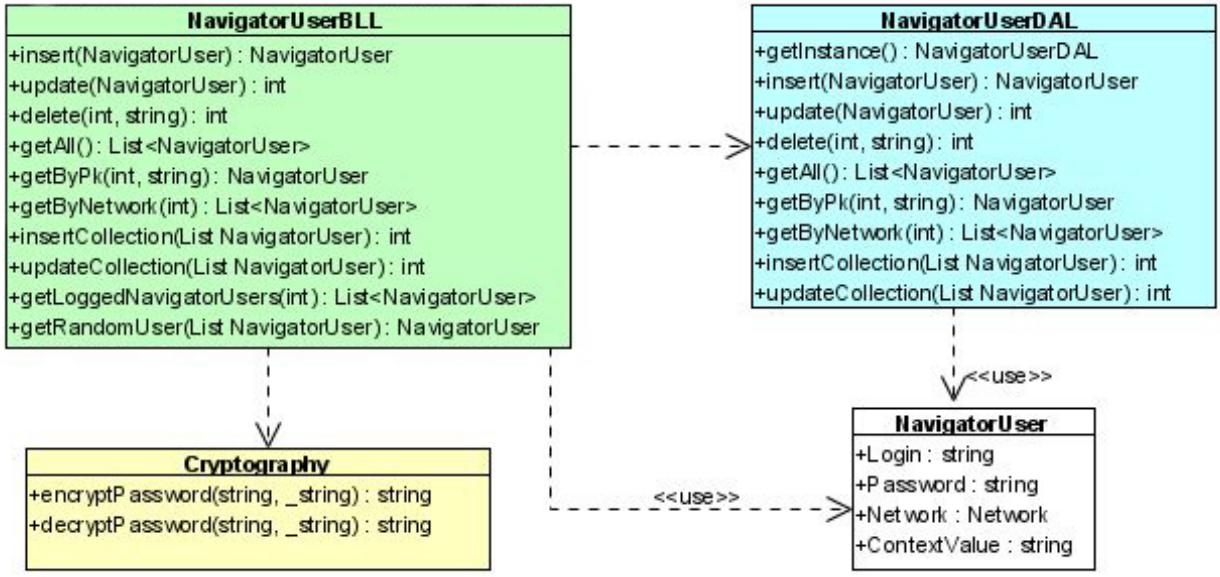


Figura 39 - Diagrama de Classes – NavigatorUser e Facade Cryptography

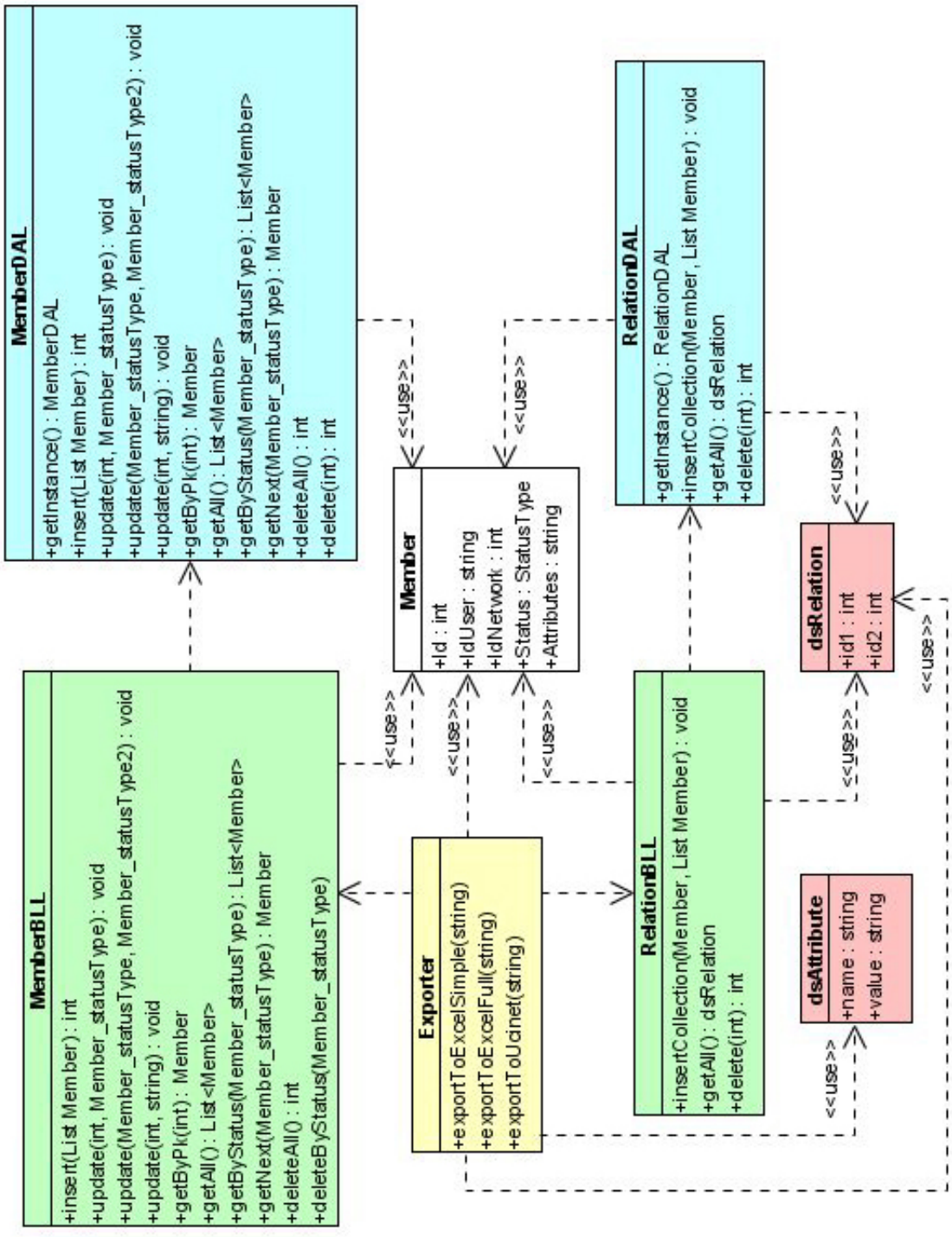


Figura 40 - Diagrama de Classes – Facade Exporter e suas interdependências

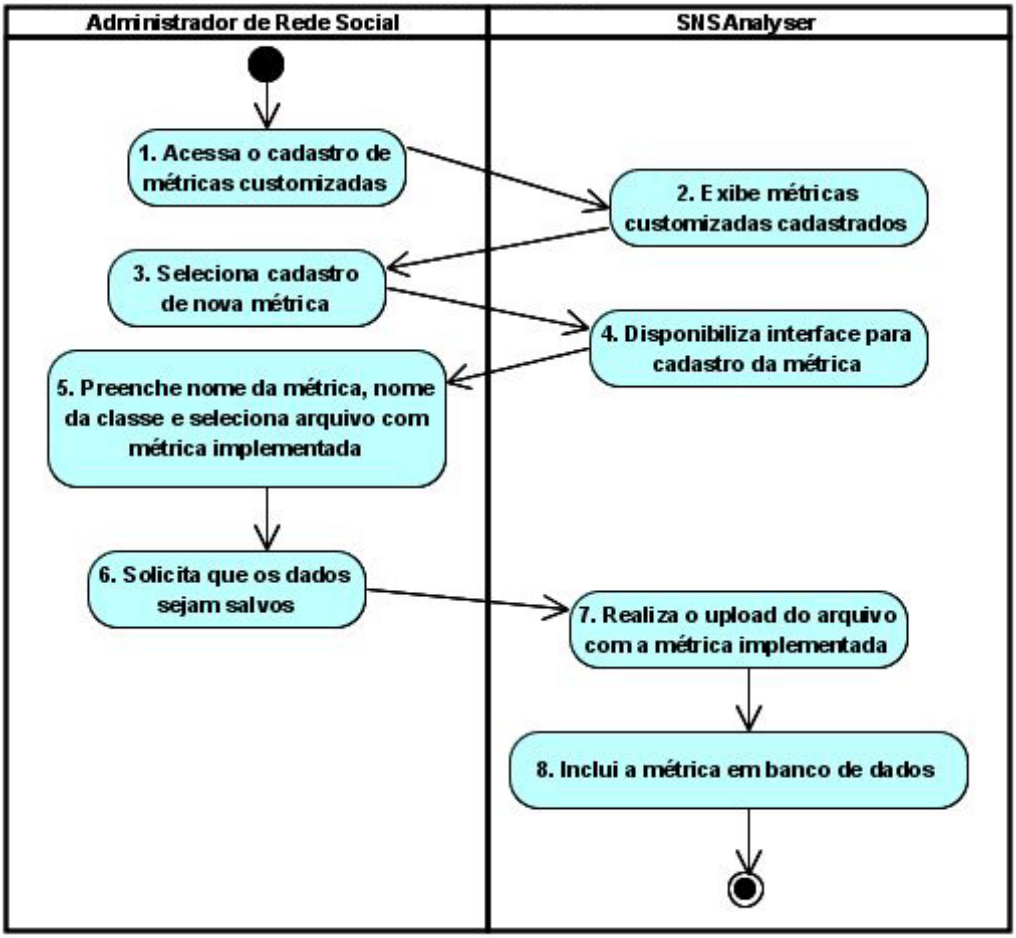


Figura 41 - Diagrama de Atividades – Incluir métrica customizada

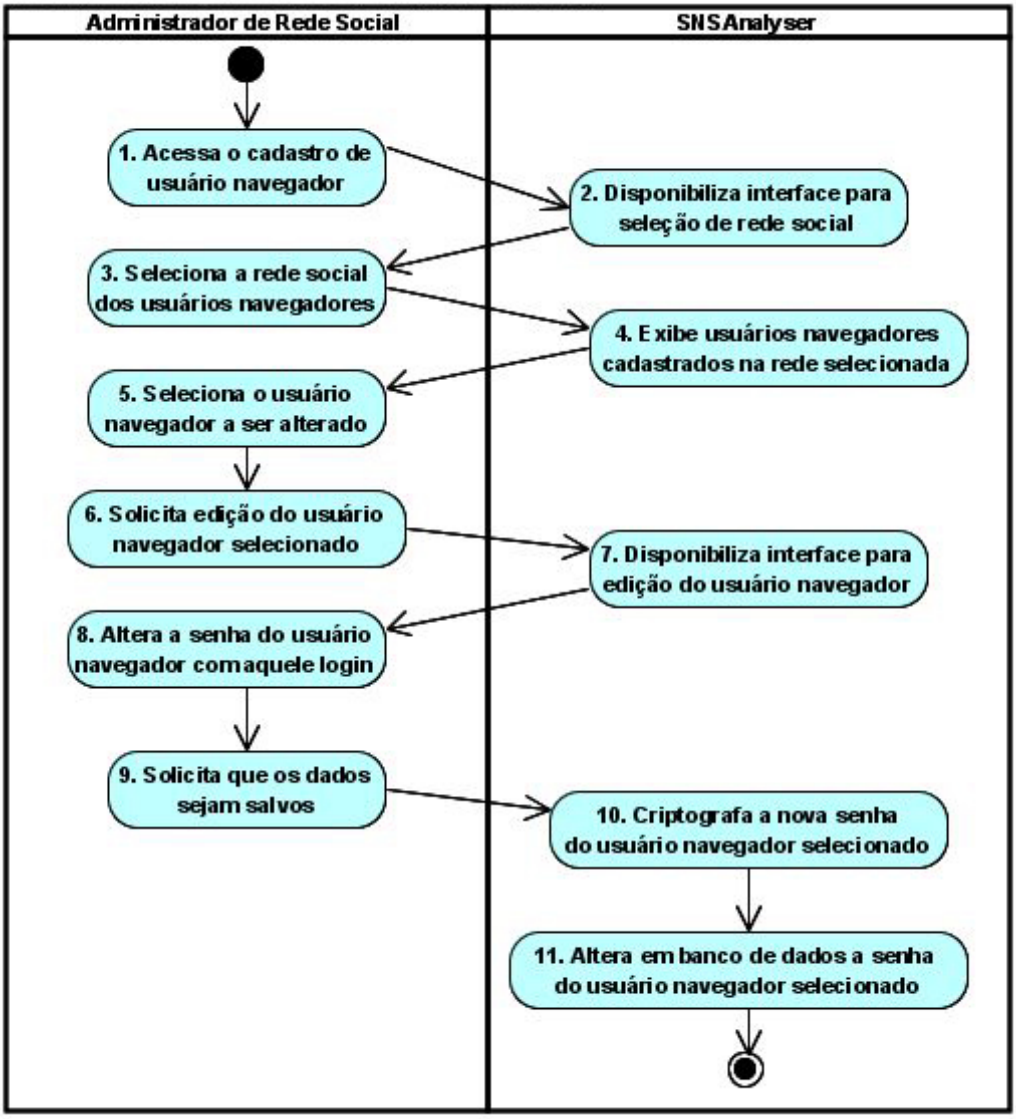


Figura 42 - Diagrama de Atividades – Alterar Usuário Navegador

APÊNDICE B – API DETALHADA DO POWER SCRIPT

1. Rule “setvar”

A regra “setvar” possibilita ao usuário definir uma ou mais variáveis dentro do script.

1.1. Variáveis

A variável é uma peça importante na definição do script, pois todos os valores extraídos são armazenados nela.

Uma variável dentro do script tem algumas características que são importantes serem observadas:

- Ao ser definida uma variável esta tem o seu escopo global e é válido até o final da execução do script.
- Toda variável criada no Script é do tipo vetor, mesmo que só tenha um elemento armazenado. Grande parte dos argumentos suporta um vetor como parâmetro, fazendo com que isso cause uma execução recursiva do comando para CADA elemento do vetor. Essa é uma característica muito importante, pois pode facilitar em muito a execução e criação script.
- O conteúdo de uma variável é sempre texto.
- Para se utilizar o conteúdo de uma variável dentro de um argumento é necessário colocar o seu nome dentro do símbolo “#”. Ex.:

```
<var name="somename" value="somevalue" />
<var name="somename2" value="#somename#" />
```

1.2. Atribuição de valores

Atributo	Req.	Valores Possíveis	Descrição
name	Sim	[Texto]	Contém o nome do variável. Se APENAS esse atributo for definido, o conteúdo do HTML obtido pela regra “read” é armazenado nesse nome. **TODO VARIÁVEL REQUER O ATRIBUTO NAME.**
value	Não	[Texto]	Define o valor que será armazenado na variável de nome “name”. Se esse atributo não for passado, automaticamente o valor assumido será o HTML obtido pela última regra READ executada.
once	Não	true false	Se “true” essa variável será definida uma única vez. Tem comportamento igual a uma constante. Mas esse atributo é particularmente útil para definir atributos externos passados pela API. É importante salientar que as definições pela API são definidas ANTES da execução dos scripts. Sendo assim com esse argumento, ele NÃO mudará o valor se este foi passado inicialmente pela API. Default: false.

1.3. Obter valores de um Atributo

Podemos extrair os atributos de um determinado elemento HTML.

Por exemplo: Para extrair todos os HyperLinks de um determinado documento HTML, podemos utilizar a seguinte declaração:

```
<var name="somename" attribute="href" of-tag="a" />
```

Atributo	Req.	Valores Possíveis	Descrição
attribute	Sim	[Texto]	Define o atributo que deseja ser extraído. Se mais de um atributo for encontrado, automaticamente será gerado um vetor.
of-tag	Sim	[Texto]	Define de qual tag que se deseja extrair um atributo.

1.4. Obter o texto de um Tag

Também é possível extrair o texto dentro de um TAG. Por exemplo:

```
<var name="somename" text-element="true" of-tag="a" />
```

Obterá um texto do tag "A" conforme listado abaixo:

```
<a href="somelink">THIS TEXT</a>
```

Atributo	Req.	Valores Possíveis	Descrição
text-element	Sim	true	Informa que se deseja extrair o texto de um determinado nó. Por exemplo: `<a href="">TEXTO</a>`
of-tag	Sim	[Texto]	Define de qual tag se deseja extrair o texto.

1.5. Obter um texto entre dois valores

É possível obter o texto entre dois valores existentes no documento HTML. É importante notar que o texto não necessariamente está dentro um TAG.

```
<var name="somename" after="sometext" before="sometext2" />
```

Atributo	Req.	Valores Possíveis	Descrição
after	Sim	[Texto]	Informa o limite inferior do texto
before	Sim	[Texto]	Informa o limite superior do texto.
include-limit	Não	true false	Se true será adicionado ao valor obtido o conteúdo de after e before. (Default: false)
revert	Não	true false	Se true o after e before será aplicado sobre o texto invertido.

1.6. Modificando o comportamento do filtro

Nas regras de extração e definição de variáveis é possível adicionar filtros que definem se o valor deve ou não ser obtido e também ações básicas que modificam o valor definido. Esses atributos podem ser combinados entre si.

Atributo	Valores Possíveis	Descrição
substring	[Inicio];[Tam]	Permite extrair um SUBSTRING começando da posição [início] e tamanho [tam]
reset	true false	Se o valor for **true** o contexto existente será apagado e um novo resultante da navegação do script será criado.

distinct	true false	Se o valor for true apenas adicionará ao vetor elementos únicos.
trim	true false	Se o valor for true retirará todos os espaços extras de cada elemento do vetor.
index	[1-10000]	Permite selecionar apenas um elemento do vetor definido por índice. O primeiro elemento começa no número 1.
add	true false	Se o valor for true adiciona elementos ao vetor ao invés de substituir.
ignore-empty	true false	Se o valor for true apenas adicionará ao vetor elementos que possuem valor.
substring-reverse	[Inicio];[Tam]	Permite extrair um SUBSTRING de trás para a frente. A posição [início] = "0" corresponde ao último caractere e tamanho[tam] corresponde quantos caracteres deverão ser obtidos.
replace	texto1|texto2	Substitui o texto1 pelo texto2. O símbolo | separa o texto 1 do texto2.
fix-amp	true false	Muda todos os símbolos & para &.
html-chars	encode decode	Possibilita fazer o HTML encode/decode do valor do vetor.
url-chars	encode decode	Possibilita fazer o URL encode/decode do valor do vetor.
contain	[Texto]	Só adiciona o valor ao vetor se este CONTIVER o texto.
not-contain	[Texto]	Só adiciona o valor ao vetor se este NÃO CONTIVER o texto.
encrypt-key	[Texto]	Permite cifrar/decifrar um texto de acordo com a chave fornecida
encrypt-url-chars	encode decode	Idem ao atributo url-chars, porém é aplicado após o encrypt-key.
encrypt-html-chars	encode decode	Idem ao atributo html-chars, porém é aplicado após o encrypt-key.
join	separador	Transforma um vetor de n posições para um de 1 posição, com os itens do vetor anterior sepados pelo SEPARADOR.
split	separador	Transforma um vetor de 1 posição para n posições, onde cada item do novo será o texto contido no intervalo entre cada ocorrência do SEPARADOR.
pad	[Tipo];[Tam]; [PaddingChar]	Preenche a string até o tamanho [Tam] desejado com o carcter informado [PaddingChar] de acordo com [Tipo], L para esquerdo ou R para Direito.

1.7. Criando uma variável do tipo Data

É possível criar variáveis que tenham informações de Data. Os atributos a seguir podem ser combinados entre si.

Atributo	Valores Possíveis	Descrição
datetime-culture	http://msdn2.microsoft. com/en-us/library/ system.globalization. cultureinfo(VS.80).aspx Seção “Culture Names and Identifiers”, coluna “Culture Name”	Cultura que será utilizada para converter o valor de entrada (atributo value) em um objeto data e para formatar o valor de saida da variável.
datetime-format-output	http://msdn2.microsoft. com/en-us/library/ system.globalization. datetimeformatinfo(VS.80).aspx	Formato que a data terá na variável de saída.
day	[Numéro]	Informa o dia que será utilizado para construir a data, caso não seja informado, será utilizado o dia da “data atual” ou o dia da data passada pelo atributo value com o formato válido de acordo com a cultura informada.
month	[Numéro]	Informa o mês que será utilizado para construir a data, caso não seja informado, será utilizado o mês da “data atual” ou o mês da data passada pelo atributo value com o formato válido de acordo com a cultura informada.
year	[Numéro]	Informa o ano que será utilizado para construir a data, caso não seja informado, será utilizado o ano da “data atual” ou o ano da data passada pelo atributo value com o formato válido de acordo com a cultura informada.
hour	[Numéro]	Informa a hora que será utilizado para construir a data, caso não seja informado, será utilizado a hora da “data atual” ou a hora da data passada pelo atributo value com o formato válido de acordo com a cultura informada.
minute	[Numéro]	Informa o minuto que será utilizado para construir a data, caso não seja informado, será utilizado o minuto da “data atual” ou o minuto da data passada pelo atributo value com o formato válido de acordo com a cultura informada.
second	[Numéro]	Informa o segundo que será utilizado para construir a data, caso não seja informado, será utilizado o segundo da “data atual” ou o segundo da data passada pelo atributo value com o formato válido de acordo com a cultura informada.
time-interval	[Numéro]	Informa o valor que será adicionado ou

		subtraído da Data.
time-interval-type	Year Month Day Hour Minute Second	Informa em que parte da data o valor informado pelo time-interval será adicionado ou subtraído

1.8. Lendo um valor do arquivo de configuração

Permite atribuir a uma variável o valor de uma chave no APP.CONFIG.

```
<var name="variavel" load-config="CHAVENOCONFIG"/>
```

1.9. Variáveis especiais

O Spider cria automaticamente as variáveis contendo os valores do Cookies. Para isso é adiciona uma variável para cada Cookie. O nome dessa variável é automaticamente definido para: "**cookie.NOMEDOCOOKIE**".

Também é criada uma variável especial de nome "**cookie.$list$**" que contém a lista de COOKIES definido nas variáveis. Essa contém os cookies separados por ;

1.10. Utilização de funções e expressões

O Script permite execute funções pré-definidas de duas formas:

- Acrescentando a cláusula evaluate="true" à variável
- Criando uma expressão entre #[e]#

Nos dois casos é permitido executar as seguintes funções:

Função	Descrição
Mod(arg1; arg2)	O resto da divisão entre arg1 e arg2
div(arg1; arg2)	O quociente da divisão entre arg1 e arg2
add(arg1; arg2)	A soma das parcelas arg1 e arg2
sub(arg1; arg2)	A subtração entre arg1 e arg2
Mult(arg1; arg2)	A multiplicação de arg1 por arg2
random(arg1; arg2)	Um número aleatório entre arg1 e arg2
len(arg1)	O tamanho do string em arg1
lenarray(arg1)	O tamanho do array representado pela variável em arg1. Note que nenhum caracter especial deve ser utilizado, apenas o nome da variável.
encode(html|url; arg2)	Faz o encode HTML ou URL de arg2
decode(html|url; arg2)	Faz o decode HTML ou URL de arg2

Exemplo:

```
<rule action="setvar">
      <var name="v" value="19" />
      <var name="v1" value="O valor é: #[ mod(13; 5) ]#" />
      <var name="v2" value="O mod #v# por 5 é: #[ mod(%v%; 5) ]#" />
      <var name="v3" value="mult(mod(13; 5); 10)" evaluate="true" />

      <var name="a" value="30" add="true" />
      <var name="a" value="45" add="true" />
```

```
        <var name="a1" value="lenarray(a)" evaluatue="true" />

        <var name="e1" value="decode(html, %v2%)" evaluatue="true" />
        <var name="e2" value="http://x/x.aspx?#[encode(url, %v1%)]#" />
</rule>
```

2. Rule “read”

Essa regra possibilita acessar uma página. O resultado dessa página ficará em um buffer que poderá ser manipulado através do tag <VAR>. A requisição por padrão é do tipo GET. Caso exista qualquer tag <POST> em qualquer parte dentro da regra RULE o sistema automaticamente faz um POST ao invés do GET. Essa regra não permite aninhar outras regras.

Ex.:

```
<!-- Exemplo de uma requisição GET -->
<rule action="read" param="http://www.orkut.com/Edit.aspx?uid=#id#">
</rule>

<!-- Exemplo de uma requisição POST -->
<rule action="read" param="http://www.orkut.com/Edit.aspx?uid=#id#">
        <post name="somename" value="somevalue" />
</rule>
```

É importante observar que no atributo “param” podem ser passadas combinações de variáveis. O Script tentará fazer uma combinação dos valores e executará um REQUEST para cada elemento do vetor resultante.

2.1. Fazendo uma requisição

Atributo	Req.	Valores Possíveis	Descrição
action	Sim	read	Determina que a ação é de leitura.
param	Sim	[url]	Determina a URL que será feita a requisição. É importante observar que após a leitura o conteúdo acessado fica disponível no Buffer para ser coletado pela diretiva <VAR>
delay	Não	[miliseconds]	Informa o tempo de espera ANTES de iniciar a requisição propriamente dita.
valid-exists	Não	[texto]	Esse atributo valida se o conteúdo da página obtido pela requisição contém o texto definido. Se não contiver o texto, um erro é levantado. Caso o atributo “code-erro” tenha sido definido o sistema levantará a mensagem de erro definida neste atributo.
valid-not-exists	Não	[texto]	Esse atributo valida se o conteúdo da página obtido pela requisição NÃO contém o texto definido. Se contiver o texto, um erro é levantado. Caso o atributo “code-erro” tenha sido definido o sistema levantará a mensagem de erro definida neste atributo.
code-erro	Não	[texto]	Mensagem de erro para os atributos valid-exists e valid-not-exists.
special-cookie	Não	true|false	Esse parâmetro só deve ser utilizado em casos

			aonde o site não consegue persistir o COOKIE de autenticação. Foi testado com sites que tem o cookie JSESSIONID do Java. Nesses casos setar para true.

2.2. Fazendo uma iteração FOR.
É possível criar um laço com início e fim pré-determinados.

Atributo	Req.	Valores Possíveis	Descrição
action	Sim	read	Determina que a ação é de leitura.
param	Sim	[url]	Determina a URL que será feita a requisição. É importante observar que após a leitura o conteúdo acessado fica disponível no Buffer para ser coletado pela diretiva <VAR>
for	Sim	[variável]	Determina a variável que irá receber o valor resultante da iteração desde START até END. Note que variável NÃO pode ser uma expressão com (#). Essa variável pode ser utilizada no parâmetro ou em qualquer regra dentro dessa regra.
start	Sim	[number]	Determina o valor inicial da iteração.
End	Não	[number]	Determina o fim da iteração. Não é obrigatório, pois esse valor pode ser definido após a primeira leitura. O nome da variável criada automaticamente é o nome da variável seguido de _end. Para definir o valor na primeira iteração a diretiva <VAR> deve conter a cláusula ONCE.
Step	Não	[number]	Determina o incremento da iteração FOR desde START até o END.

2.3. A declaração Post
Ao definir um “post” dentro de uma regra “read” automaticamente o método da requisição é chaveado para POST.

```
<post name="aaa" value="aaa" />
```

Atributo	Req.	Valores Possíveis	Descrição
Name	Sim	[Texto]	Contém o nome do parâmetro POST.
Value	Sim	[Texto]	Contém o valor a ser enviado.
index	Não	[1-10000]	Se for passado um valor através do INDEX, o post será apenas do elemento referenciado pela posição. Caso contrário o POST será feito com TODOS os elementos do vetor.
not-empty	Não	True False	Se true, só criará uma entrada no POST se o elemento não estiver vazio. O default é sempre postar qualquer elemento.

2.4. Enviando um Cookie
É possível enviar um cookie para uma requisição WEB.

```
<cookie name="nomecookie" value="valorcookie" />
```

Atributo	Req.	Valores Possíveis	Descrição
Name	Sim	[Texto]	Contém o nome do Cookie
Value	Sim	[Texto]	Contém o valor do Cookie.
domain	Não	[Texto]	O domínio do cookie.
path	Não	[Texto]	O path do Cookie.

2.5. Validação do conteúdo

Além da validação na definição da regra é possível adicionar uma diretiva chamada <validation>. Ela precisa ser adicionada dentro da cláusula <rule action=”read”>. O grande diferencial é que essa cláusula suporta EXPRESSÕES (substituições de variáveis) enquanto que a cláusula dentro da definição da regra valida o documento inteiro.

```
<!-- A expressão #AUX# não deve ter "forgotpasswd" -->
<validation value="#aux#" not-exists="forgotpasswd" code-erro="5002"/>

<!-- A expressão #AUX# deve ter pageerror -->
<validation value="#aux#" exists="pagaerror" code-erro="5003"/>
```

2.6. Modificando o cabeçalho da requisição

É possível modificar o cabeçalho da requisição adicionando o seguinte código dentro da cláusula <rule action=”read”>

```
      <httpheader>
            <header name="" value="" />
            <header name="" value="" />
            <header name="" value="" />
      </httpheader>
```

3. Rule “database”

Através dessa regra é possível interagir com o banco de dados executando qualquer script DML. Inicialmente, é necessário setar no arquivo de configuração (app.config) da aplicação que irá rodar o spider as seguintes opções:

```
<!--DATABASE CONNECTION
  Object (including namespace) to connect to database
  Example: System.Data.SqlClient.SqlConnection -->
<add key="spider.DATABASECONNECTION"
value="System.Data.SqlClient.SqlConnection" />

<!--DATABASE ASSEMBLY
  Assembly name (without .dll) it contains Object Connection
  Example: System.Data -->
<add key="spider.DATABASEASSEMBLY" value="System.Data" />

<!--CONNECTION STRING name represents the name of connection string.-->
<add key="spider.CONNECTIONSTRING.name"
value="Server=name;Database=name;User Id=xyz;Password=" />
```

A cláusula <database> define a conexão e o tipo de transação e as cláusulas <query> definem o tipo de consulta que deverá ser feito.

3.1. A definição da conexão dentro do script

Atributo	Req.	Valores Possíveis	Descrição
action	Sim	[Texto]	Nome da ação: “database”

connection	Sim	[Texto]	Indica qual a o nome da conexão que as queries irão utilizar. Os nomes de conexões estão definidos dentro do arquivo de configuração (app.config)
transaction	Não	true false	Se true, indica que a execução de todas as queries serão transacionais.

3.2. Executando consultas

Existe 5 tipos diferentes de consultas: texto livre, escalar, dinâmica inteligência, dinâmica update, dinâmica delete. Essa regra não permite aninhar outras regras.

```
<rule action="database" connection="string" transaction="true" >
  <!-- Esse tipo de consulta define um SQL Livre -->
  <query sql="text" index="1">
    insert into tabela (campo1, campo2) values ('#var#', '#var2#');
  </query>
  <!-- Esse tipo de consulta define um SQL Livre mas que o retono
      deve obrigatoriamente ser UM VALOR apenas. Esse valor será
      armazenado em "variável".
  -->
  <query sql="scalar" index="1" result="variavel">
    select count(*) from tabela
  </query>
  <!--
  ATENCAO: O Execute-on consome muitos recursos de máquina.
     usar apenas se necessário.
  -->
  <query sql="dynupdate" table="name" index="1" execute-on="expression"
sql-where=" expressao " >
    <field name="field" key="true" value="#var#" default="NULL"
type="string|number|date" />
    <field name="field" value="#var#" default="NULL"
type="string|number|date" />
  </query>
  <query sql="dyninsert" table="name" index="1" execute-on="expression" >
    <field name="field" key="true" value="#var#" default="NULL"
type="string|number|date" />
    <field name="field" value="#var#" default="NULL"
type="string|number|date" />
  </query>
  <query sql="dynintelligent" table="name" index="1" execute-
on="expression" sql-where=" expressao " >
    <field name="fieldkey" key="true" value="#var#" default="NULL"
type="string|number|date" />
    <field name="field" value="#var#" default="NULL"
type="string|number|date" depend-table="xxx" depend-field="zzz" />
  </query>
</rule>
```

4. Rule "savefile"

Essa regra possibilita salvar o conteúdo de uma URL ou o conteúdo de uma variável no sistema de arquivos. Essa regra não permite aninhar outras regras.

Atributo	Req.	Valores Possíveis	Descrição
action	Sim	[Texto]	Nome da ação: "savefile"
url	Não	[Texto]	Vetor que Indica a URL dos arquivos que serão salvos no disco. OBSERVAÇÃO: Se esse parâmetro não for

			passado a regra automaticamente entende que é para se trabalho o InnerXml da expressão entre o <rule></rule>
path	Não	[Texto]	Indica o caminho padrão a salvar os documentos. Se omitido utiliza o caminho atual ou o FULLNAME definido em "filename"
filename	Não	[Texto]	Indica o nome do documento. Se omitido assume o nome completo da URL.
overwrite	Não	true false	Se for definido como TRUE sobrescreve o arquivo existente, se definido como FALSE , adiciona um novo arquivo com um contador.

5. Rule "email"

Essa regra possibilita mandar um email através do Script. A mensagem a ser enviada está dentro do InnerXml da regra "email". Essa regra não permite aninhar outras regras.

Atributo	Req.	Valores Possíveis	Descrição
action	Sim	[Texto]	Nome da ação: "email"
smtp	Sim	[Texto]	Informa o nome do servidor SMTP que irá enviar as mensagens.
authuser	Não	[Texto]	Se o servidor SMTP requerer autenticação deve ser passado o usuário.
authpass	Não	[Texto]	Se o servidor SMTP requerer autenticação deve ser passado o usuário e senha.
from	Sim	[Texto]	Email do remetente
to	Sim	[Texto]	Email do destinatário. Suporta vetor.
subject	Sim	[Texto]	Assunto do email.

6. Rule "delay"

Essa regra provoca uma pausa na execução do script pelo tempo informado. Essa regra não permite aninhar outras regras.

Atributo	Req.	Valores Possíveis	Descrição
action	Sim	[Texto]	Nome da ação: "delay"
time	Sim	[milisecs]	Tempo em milisegundos da parada.

7. Rule "executeif"

Essa regra limita a execução de um determinado bloco apenas se uma determinada condição for satisfeita. A condição e baseada em um resultado de uma expressão comparado se é vazio, não vazio, igual ou não a igual a valor.

Atributo	Req.	Valores Possíveis	Descrição
action	Sim	[Texto]	Nome da ação: "executeif"
cond	Sim	[Expressão]	O resultado de alguma variável
type	Sim	equal not-equal empty	O tipo de comparação que será feito. OBS.: Caso o cond e o value sejam valores

		not-empty less-than less-equal-than greater-than greater-equal-than	NUMÉRICO o less-than, less-equal-than, greater-than, greater-equal-than farão comparação numérico para numérico, caso contrário, farão comparação string para string.
value	Não	[texto]	Esse campo só é utilizado quando o tipo de comparação for equal ou not-equal

8. Rule "for-each"
Essa regra possibilita percorrer CADA elemento de uma variável do tipo vetor e gerar uma repetição com esses valores.

Atributo	Req.	Valores Possíveis	Descrição
action	Sim	[Texto]	Nome da ação: "for-each"
collection	Sim	#var#	A expressão que contém o vetor.
item	Sim	[variável]	Nome da variável que receberá o elemento unitário referente a iteração em collection.
indexvar	Não	[variável]	Nome da variável que irá contar quantas iterações foram executadas.
collection-length-var	Não	[variável]	Nome da variável que irá conter o tamanho da coleção em questão.

9. Rule "For"
A regra "for" possibilita fazer uma iteração um determinado número conhecido de vezes.

Atributo	Req.	Valores Possíveis	Descrição
action	Sim	[Texto]	Nome da ação: "for"
var	Sim	[Texto]	Nome da variável que será criada para a repetição
start	Sim	[Inteiro]	Primeiro valor da repetição a ser atribuído em "var"
end	Sim	[Inteiro]	Último valor da repetição a ser atribuído em "var"
step	Não	[Inteiro]	Incremento da variável "var"

```
<rule action="for" var="i" start="1" end="10" step="1">
      <rule ......>
      </rule>
</rule>
```

10. Rule "Throw"
Permite levantar a exceção com uma mensagem específica.
```
<rule action="throw" message="mensagem da exceção" />
```

11. Rule "Call" (Procedures)
Com o PowerScript é possível organizar o código em procedures para fazer a reutilização de código. Nesse caso, basta chamar a regra "call" e informar o nome da procedure, conforme exemplo abaixo.

Uma característica interessante é que o parâmetro procedure dessa regra aceita um vetor, sendo assim podem ser múltiplas chamadas em uma única regra.

Observação: O nó procedure deve obrigatoriamente estar dentro de “processpage” para poder ser encontrado.

```
<processpage>
      <context name="to-be-set" />
      <rules>
            .
            .
            <rule action="call" procedure="nome_procedure" />
            .
            .
      </rules>

      <procedure name="nome_procedure">
            <rule ... >
            </rule>
      </procedure>
</processpage>
```

12. Rule “xml”

Está regra permite a criação de variáveis do tipo xml, para isso basta seguir o exemplo apresentado abaixo:

```
<processpage>
  <context name="not-be-set" />
  <rules>
    ...
    <rule action="xml" xml-output-var="variavel" version="versão"
encondig="enconding" standalone="standalone">
      <element name="nome_elemento_1">
        <attribute name="nome_atributo" value="valor atributo"/>
        <element name="nome_elemento_2">
          <element name="nome_elemento_3" value="valor 3" />
          <rule action="for-each" ... >
            <rule action="xml" xml-output-var="variavel">
              <element name="nome_elemento_4">
                <element name="nome_elemento_5" value="valor 5"/>
              </element>
            </rule>
          </rule>
        </element>
      </element>
    </rule>
  </rules>
</processpage>
```

Como saída em “variavel” temos:

```
<?xml version="versão" encoding="encoding" standalone="standalone" ?>
<nome_elemento_1 nome_atributo="valor atributo">
  <nome_elemento_2>
    <nome_elemento_3>valor 3</nome_elemento_3>
    <nome_elemento_4>
      <nome_elemento_5>valor 5</nome_elemento_5>
    </nome_elemento_4>
    <nome_elemento_4>
      <nome_elemento_5>valor 5</nome_elemento_5>
    </nome_elemento_4>
  </nome_elemento_2>
</nome_elemento_1>
```

Atributo	Req.	Valores Possíveis	Descrição
version	Não	[Texto]	Versão do xml a ser criado, se não for informado o sistema assuma “1.0”
enconding	Não	[Texto]	Encoding que será usado pelo xml a ser criado, se não for informado assuma “utf-8”.
standalone	Não	yes no	Informa se o xml é ou não standalone, se não for informado o xml não utiliza essa propriedade.

13. Tratamento de Erro (Catch)

Cada regra “RULE” e as suas regras aninhadas podem ter o seu bloco protegido por uma ou mais cláusulas “catch”.

Essas cláusulas evitam que o erro seja passado para o usuário. O princípio básico é tratar erros gerados por validação de código (validation, valid-exists, valid-not-exists, etc), mas também pode tratar outros erros através de suas mensagens.

É importante observar que o tratamento de erro é válido APENAS o bloco dentro do “RULE READ”. As demais regras não são protegidas.

```
<processpage>
      <context name="to-be-set" />
      <rules>
            <rule action="read" param="http://www.uol.com.br">
                  <validation code-erro="1000" valid-exists="xx" />
                  .
                  .
                  .
                  <catch code-erro="1000">
                        <rule>
                        </rule>
                  </catch>
            </rule>
      </rules>
</processpage>
```