

UNIVERSIDADE PRESBITERIANA MACKENZIE
MESTRADO EM ENGENHARIA ELÉTRICA

Evandro Luquini

**UMA PROPOSTA PARA PROMOVER A APRENDIZAGEM NAS
DISCIPLINAS DE PROGRAMAÇÃO UTILIZANDO-SE DE REDES
SOCIAIS MODELADAS POR TÉCNICAS DE DETECÇÃO DE PLÁGIO**

São Paulo
2010

UNIVERSIDADE PRESBITERIANA MACKENZIE
MESTRADO EM ENGENHARIA ELÉTRICA

Evandro Luquini

**UMA PROPOSTA PARA PROMOVER A APRENDIZAGEM NAS
DISCIPLINAS DE PROGRAMAÇÃO UTILIZANDO-SE DE REDES
SOCIAIS MODELADAS POR TÉCNICAS DE DETECÇÃO DE PLÁGIO**

Dissertação apresentada como requisito para a obtenção do Título de Mestre no Programa de Pós-Graduação em Engenharia Elétrica, área de concentração em Engenharia de Computação, da Universidade Presbiteriana Mackenzie.

Orientador: Prof. Dr. Nizam Omar

São Paulo
2010

L966p Luquini, Evandro.

Uma proposta para promover a aprendizagem nas disciplinas de programação utilizando-se de redes sociais modeladas por técnicas de detecção de plágio / Evandro Luquini – 2010 .

149 f. : il. ; 30 cm

Dissertação (Mestrado em Engenharia Elétrica) – Universidade Presbiteriana Mackenzie, São Paulo, 2010.

Bibliografia: f. 75-86.

1. Plágio. 2. Algoritmos de detecção de plágio. 3. Modelagem de redes sociais. 4. Ensino-aprendizagem de programação. I. Título.

CDD 621.30285

Evandro Luquini

**UMA PROPOSTA PARA PROMOVER A APRENDIZAGEM NAS
DISCIPLINAS DE PROGRAMAÇÃO UTILIZANDO-SE DE REDES
SOCIAIS MODELADAS POR TÉCNICAS DE DETECÇÃO DE PLÁGIO.**

Dissertação apresentada como requisito para a obtenção do Título de Mestre no Programa de Pós-Graduação em Engenharia Elétrica, área de concentração em Engenharia de Computação, da Universidade Presbiteriana Mackenzie.

Aprovada em agosto de 2010

BANCA EXAMINADORA

Prof. Dr. Nizam Omar

Universidade Presbiteriana Mackenzie - UPM

Prof. Dr. Leandro Augusto da Silva

Universidade Presbiteriana Mackenzie - UPM

Prof. Dr. Guiou Kobayashi

Universidade Federal do ABC - UFABC

AGRADECIMENTOS

Em especial agradeço ao meu orientador prof. Dr. Nizam Omar pelo direcionamento, intervenção e apoio necessários ao amadurecimento desta pesquisa, da minha visão sobre o trabalho científico e o universo acadêmico.

A minha esposa, amigos e familiares pela compreensão, apoio e críticas sempre que oportuno.

Enfatizo meu apreço e reforço meus agradecimentos à CAPES e ao Instituto Presbiteriano Mackenzie pela bolsa PROSUP que me foi concedida.

RESUMO

Este trabalho está inserido no contexto das pesquisas que procuram desenvolver ambientes computacionais que apóiam efetivamente os processos de ensino aprendizagem. Em especial, devido à grande demanda da sociedade por recursos humanos capazes de construir e manter sistemas computacionais. O tema específico desta pesquisa concentra-se na aplicação de técnicas de redes sociais e algoritmos de detecção de plágio à questão da aprendizagem de programação. Atualmente os professores envolvidos com o ensino de programação possuem um conjunto de ferramentas capazes de detectar o plágio de código-fonte nos exercícios e avaliações realizados por seus alunos. Apesar de estas ferramentas serem úteis como instrumentos disciplinadores, elas não permitem ao docente refletir sobre a natureza social do plágio. Este trabalho desenvolve e instrumentaliza a hipótese de que o plágio, no contexto do ensino de programação, não acontece isoladamente da rede social formada pelos alunos em sala de aula. Em decorrência desta formulação propõe-se que os algoritmos de detecção de plágio e os códigos fonte apresentados durante uma disciplina serão suficientes para modelar indiretamente a rede social estabelecida pelo grupo de alunos. Um experimento exploratório foi conduzido para avaliar esta hipótese e um procedimento para intervenção inspirado nas técnicas de imunização de redes sociais foi proposto com o intuito de aumentar a eficácia da aprendizagem.

Palavras-chave: *plágio, algoritmos de detecção de plágio, modelagem de redes sociais, ensino-aprendizagem de programação.*

ABSTRACT

The broad topic of this research is related to computer supported learning and teaching environment. In particular due to the high society's demand for human resources capable of building and maintaining computer systems the specific topic of this research is focused on computer tools to support programming learning and teaching using techniques from social network and plagiarism detection. Currently the teachers involved with programming teaching have a set of tools to detect plagiarism of source code in the exercises and assessments made by their students. Although these tools are useful as disciplinary measures they do not allow teachers to think about plagiarism from a social perspective. This paper develops and exploits the assumption that plagiarism in the context of programming teaching does not happen in isolation from the social network formed by students in the classroom. From this formulation is suggested that the plagiarism detection algorithms and the students source code presented during the course are sufficient to model indirectly the social network established by the group of students. An exploratory experiment was conducted to evaluate this hypothesis and an intervention procedure was proposed. This procedure was inspired by immunization techniques from social networks and it has the goal to improve learning.

Key Words: *Programming plagiarism, social network analysis, programming teaching, plagiarism detection.*

LISTA DE FIGURAS

Figura 1 - Programa original à esquerda e programa alterado a direita (LIU et. al, 2006)	25
Figura 2 - Visualização de similaridades (KOSCHKE, 2008)	29
Figura 3 - Abstração Léxica do Código Java (PRCHELT, 2002).....	30
Figura 4 - Abstração Léxica dos Identificadores (PRECHELT, 2002)	30
Figura 5 - Abstração Sintática de um Procedimento. À esquerda o código a direita sua representação sintática.....	31
Figura 6 - Exemplo de Grossman e Frieder (2004) para o procedimento LSI.....	35
Figura 7 - Exemplo de Grossman e Frieder (2004) para o procedimento LSI com fator de correção	36
Figura 8 - Visualização gráfica de Kontostathis (2006) para o método LSI.....	37
Figura 9 - Redução usando k igual a 2.....	38
Figura 10 - Comparação entre as matrizes com e sem redução.....	38
Figura 11 - Exemplo de Redução do vetor	39
Figura 12 - Exemplo de um grafo construído a partir de um programa (LIU et. al, 2006)	40
Figura 13 - Exemplo de relatório de detecção de plágio do JPLAG (PRECHELT, 2002)	43
Figura 14 - Ferramenta de visualização AC (FREIRE, 2008).....	43
Figura 15 - Uma nova interpretação para as ocorrências de plágio no ensino de programação (LUQUINI e OMAR, 2009).....	44
Figura 16 - Apresentação gráfica da rede social (SIH, 2009).....	45
Figura 17 - Apresentação matricial da rede social (CHURCHILL,2005).....	46
Figura 18 - Imunização dos alunos fornecedores (nós envoltos no retângulo) inspirado na metáfora de fragmentação da rede em processos de combate a epidemias	49
Figura 19 - Grafo sociométrico proveniente do questionário aplicado aos alunos	54
Figura 20 - Exemplo de decomposição do programa em sua forma algébrica.....	58
Figura 21 - Comparação dos trabalhos do aluno 12 no exercício 11 com os outros alunos usando dimensão k=10 e variando o fator de Entropia.....	59

Figura 22 - Comparação visual entre os trabalhos dos alunos 12 e 44 no exercício 11: Usando o fator de entropia os documentos são apontados como semelhantes.	60
Figura 23 - Comparação visual entre os trabalhos dos alunos 12 e 30 no exercício 11: Usando o fator de entropia os documentos são apontados como semelhantes.	60
Figura 24 - À esquerda distribuição com $k = 2$ e a direita com $k = 5$	61
Figura 25 - À esquerda distribuição com $k = 10$ e a direita com $k = 15$	62
Figura 26 - À esquerda distribuição com $k = 20$ e a direita com $k = 25$	62
Figura 27 – Matrizes representando dois tipos de relacionamentos distintos	64
Figura 28 – Distribuição das Correlações	68
Figura 29 – Desvio Padrão das Correlações	68
Figura 30 – Rede do Ego do Aluno 2	72
Figura 31 – Associação entre os alunos 2 e 21	73
Figura 32 – Associação entre os alunos 2, 31 e 21	73
Figura 33 – Associação entre os alunos 2, 21 e 1	74

LISTA DE TABELAS

Tabela 1 - Propriedades de Centralidade e Poder (CHURCHILL,2005)	47
Tabela 2 – Disposição da entrega dos trabalhos	53
Tabela 3 – Distribuição de Similaridades	64
Tabela 4 – Resultado do QAP para todos os trabalhos	66
Tabela 5 – Resultado do QAP para todos os trabalhos	67
Tabela 6 – Comparação das relações entre os trabalhos isoladamente	71

SUMÁRIO

1.	INTRODUÇÃO	13
1.1	JUSTIFICATIVA	15
1.2	HIPÓTESES.....	18
1.3	ORGANIZAÇÃO.....	19
2.	REFERENCIAL TEÓRICO.....	20
2.1	A RELEVÂNCIA DA HABILIDADE DE PROGRAMAR.....	20
2.2	O ENSINO E A APRENDIZAGEM DE PROGRAMAÇÃO	21
2.3	O PLÁGIO DE CÓDIGO FONTE	23
2.4	TECNOLOGIAS DE DETECÇÃO DE PLÁGIO.....	27
2.4.1	Detecção por comparação textual	28
2.4.2	Detecção por comparação léxica.....	29
2.4.3	Detecção por Árvores Sintáticas	31
2.4.4	Detecção por Métricas	32
2.4.4.1	Detecção por Métricas LSI	33
2.4.5	Detecção por Grafos	40
2.4.6	Detecção por Estilometria	41
2.5	OS DETECTORES E AS REDES SOCIAIS	42
2.6	AS REDES SOCIAIS.....	45
2.7	AS REDES SOCIAIS E A COOPERAÇÃO.....	48
3.	EXPERIMENTO	50
3.1	A METODOLOGIA	50
3.2	A POPULAÇÃO.....	51
3.3	A DISCIPLINA	51
3.4	OS ARTEFATOS.....	51
3.5	A FUNÇÃO PLÁGIO.....	55
3.6	PRÉ- PROCESSAMENTO	56
3.7	O FATOR DE AJUSTE	59
3.8	ESCOLHA DA DIMENSÃO	61
3.9	CÁLCULO DO QAP.....	63
3.10	RESULTADOS	68

	12
4. CONCLUSÃO	70
4.1 TRABALHOS FUTUROS.....	70
REFERÊNCIAS BIBLIOGRÁFICAS	75
ANEXOS	87

1. INTRODUÇÃO

A habilidade de programar é uma das competências fundamentais que devem ser adquiridas pelos egressos nos cursos de computação ou áreas relacionadas como sistemas de informação e engenharia de software (BILLS, 2007; HEMMENDINGER, 2007). Conforme registrado na literatura, o desenvolvimento pleno da competência de programar somente pode ser alcançado caso o aluno receba apropriada motivação e oportunidades para praticá-la. Através da prática o aluno exercita e desenvolve sua habilidade em utilizar um dispositivo computacional e ferramentas para elaborar um algoritmo que solucione um problema proposto (WINSLOW, 1996; CHEANG et. al., 2003; GARCIA-MATEOS, 2009). Por este motivo varias atividades desta natureza são propostas ao longo das disciplinas de programação e são avaliadas pelos professores através de uma abordagem formativa ou somativa. Em virtude da importância da prática para o desenvolvimento da competência de programar, qualquer ação de plágio promovida pelo aluno é uma grande ameaça ao processo de aprendizagem com sérias repercussões para a comunidade de computação e para a imagem das instituições de ensino (DICK et. al., 2003; DEY, 2006).

Na tentativa de evitar a problemática do plágio e as suas conseqüências, algumas instituições adotam a prática de programação somente como um instrumento formativo. Esta abordagem reforça que em função da dificuldade de controlar as possíveis ações de plágio e de mensurar o grau de dependência do trabalho apresentado pelo indivíduo em relação ao grupo, o aluno não pode ser avaliado adequadamente através dos processos e dos produtos da experiência prática de programação. Neste caso as atividades práticas têm pouca ou nenhuma relevância direta na avaliação final dos alunos em detrimento das avaliações tradicionais como provas descritivas, de múltiplas escolhas ou exames de curta duração realizados em computadores do laboratório (HWANG, 1982; JACOBSON, 2000; MCCRACKEN et. al., 2001; BARROS et. al., 2003; LISTER, 2003; DALY, 2004; HIGGINS, 2005; BENNEDSEN, 2006; YADIN, 2008). Conforme a importância da prática no critério de avaliação diminui é esperado um menor interesse por esta atividade. Conseqüentemente o aluno perderá a oportunidade de desenvolver

plenamente tal competência. Os registros que mostram alunos capazes de passar em testes tradicionais sem efetivamente serem capazes de operacionalizar o conhecimento adquirido (LUJAN, 2006) reforçam o risco de tal estratégia que aborda a prática em caráter exclusivamente formativo.

As instituições que adotam a prática de programação em caráter somativo reforçam aos seus alunos a importância desta atividade e reconhecem que os testes tradicionais não são capazes de mensurar a habilidade de programar que foi desenvolvida pelo aluno. Neste caso o plágio toma uma dimensão ainda maior. Nas situações onde a atividade prática é realizada através de uma abordagem somativa, quando um aluno plagia todo ou parte do trabalho de terceiros e apresenta-o como seu, este está ganhando crédito por uma habilidade que não possui, ou por um esforço que não fez. O que geralmente sugere que este aluno também não está apto para as próximas etapas do currículo acadêmico.

Devido aos efeitos negativos e a dificuldade do professor detectar manualmente o plágio, muita atenção é dedicada pela comunidade de computação à construção de ferramentas capazes de identificá-lo automaticamente nos códigos fontes apresentados pelos alunos (ROSALES et. al., 2008). Estas ferramentas apresentam a vantagem de serem capazes de pesquisar um grande número de repositórios e fazerem comparações e associações que não poderiam ser feitas facilmente por operações manuais. As ferramentas trabalham no contexto estrutural dos programas, buscam padrões e utilizam heurísticas derivadas das estratégias mais comuns empregadas pelos estudantes. Contudo tais ferramentas são limitadas no tratamento das questões semânticas do plágio, o que é reconhecidamente um problema de decisão computacional (BERZAL, 2007). No estado atual da técnica, as ferramentas são capazes de propor um índice de similaridade ou apontar as partes do artefato que foram copiados integralmente, deixando a cargo do professor a responsabilidade pela análise final, justificativa e decisão sobre as penalidades apropriadas.

Apesar das ferramentas de detecção de plágio contribuir para disciplinar os estudantes e inibir as práticas de plágio, os professores possuem poucos incentivos para utilizá-las. Provar o plágio e punir os estudantes geralmente é um processo que requer esforço extra do professor e implica em riscos para a sua imagem diante dos

seus pares (HARRIS, 1994; JOY, 1999; SRAKA, 2009). Invariavelmente sua qualidade de professor é colocada em dúvida como parte da estratégia de defesa do aluno. Além disto, a quantidade anormal de reprovações nas disciplinas de programação, o aumento da proporção de alunos por professor e a evasão crescente principalmente nos bacharelados em computação sugerem que exista maior pressão sobre os professores para relaxar o tratamento do plágio nas atividades práticas de programação (BENNEDSEN, 2007; CASSEL et. Al, 2007; SPINELLIS, 2007).

Os cenários apresentados reforçam a percepção de alguns autores de que a academia não está tendo sucesso no controle do plágio (SHEARD, 2003; RUDOLPH, 2007). Estes autores acreditam que uma solução melhor pode ser alcançada se uma maior ênfase de pesquisa for direcionada para o desenvolvimento de ambientes de aprendizagem mais motivadores e ações preventivas. Esta recomendação contrasta com a abordagem atual contra o plágio que intensifica o antagonismo entre alunos e professores, pouco colaboram como instrumentos de prevenção e dificultam a discussão do plágio em sala de aula como um problema de aprendizagem. A proposta deste trabalho enfatiza uma abordagem diferenciada no tratamento do plágio em sala de aula. Ao contrário de ações atuais, esta pesquisa modela o plágio como um evento social e descreve como esta ação pode ser utilizada para promover a interação e a aprendizagem entre os alunos.

1.1 JUSTIFICATIVA

Conforme sugerido por Luquini e Omar (2009), uma vez que as atividades práticas sejam concebidas de forma que não possam ser plagiadas de fontes externas, o aluno disposto ao plágio irá recorrer aos seus colegas de classe. A escolha do colega não acontecerá aleatoriamente, mas será feita com base na rede social do aluno. Uma rede social é uma descrição dos relacionamentos entre os indivíduos de um grupo, sua natureza e a intensidade destes relacionamentos. É conhecido da análise de redes sociais que a topologia da rede social oferece oportunidades e impõe restrições sobre os indivíduos do grupo. Essas oportunidades e restrições controlam a transferência dos recursos (materiais ou

imateriais) que circulam na rede, conseqüentemente determinando quando, quais e para quem os recursos estarão disponíveis (MILKA, 2007).

Neste trabalho o problema do plágio foi reinterpretado através da metáfora da transferência de recursos na rede social. No caso específico do ensino-aprendizagem de programação, os recursos envolvidos são os códigos fontes dos exercícios ou testes de programação que os alunos precisam ter para serem aprovados, e a rede social é formada pelos relacionamentos entre os próprios alunos. Esta proposição tem o mérito de formalizar a declaração de Deibel (2005) de que existe uma tendência natural dos alunos em agruparem-se ao redor dos colegas com mais afinidades e expande o conceito de cadeias de plágio percebidas por Daly (2005) durante o seu experimento.

Conforme apontado por Deibel (2005), a maneira espontânea dos alunos agruparem-se em sala de aula não é provavelmente a melhor opção para produzir o fenômeno da aprendizagem cooperativa. O autor sugere que as preferências naturais por determinados grupos podem resultar em uma relação assimétrica de aprendizagem. Esta assimetria de aprendizagem seria ocultada pelo vínculo entre os alunos e teria grande probabilidade de resultar em plágio. O trabalho de Daly (2005) também colabora com esta percepção de assimetria quando relata que os alunos identificados como fornecedores de plágio obtêm resultados melhores nos testes do que os alunos que são simplesmente receptores. No seu experimento o autor conjectura que o plágio tem resultados positivos para os alunos fornecedores e elabora que provavelmente este comportamento aumentaria a auto-estima do indivíduo reforçando a continuidade do suporte ao plágio dos seus colegas.

O reconhecimento de que os alunos estão associados a uma sala de aula através de sua rede social e assumindo que esta rede social formada espontaneamente pode não corresponder à rede ideal para a aprendizagem dos alunos, propõe-se um método capaz de realizar intervenções indiretas ao longo da evolução da disciplina de programação de forma a provocar o surgimento de novas interações sociais que possam promover a aprendizagem. Este conceito está alinhado com as hipóteses de que a aprendizagem individual é reforçada pela interação social (VYGOTSKY, 1978; JOHNSON E JOHNSON, 2002). Além disto, é esperado que a intervenção exponha naturalmente os alunos com deficiências de

aprendizagem e também que atue como um inibidor do plágio.

O procedimento idealizado assume que é possível computar a rede social dos alunos através dos artefatos que são produzidos regularmente durante as disciplinas de programação. No momento em que a rede social esteja modelada através da aplicação dos detectores de plágio nos códigos fontes (LUQUINI e OMAR, 2010), seria possível determinar quais são os indivíduos específicos que deverão receber a intervenção. Inspirado nos trabalhos de imunização de redes sociais que são utilizados para prevenir a propagação de rumores, pânico e epidemias (KLEINBERG, 2008), o procedimento aplicará o algoritmo de imunização derivado do problema do nó crítico (ARULSELVAN, 2008) para determinar quais são os alunos que precisam receber atividades diferenciadas, porém de mesma complexidade, para que estes não tenham oportunidade de propagar seu código fonte entre seus colegas. Por analogia com o controle de epidemia, a inibição do contágio é dada pela sobrecarga do nó, no caso o aluno, com uma atividade que concentre suas energias e o distancie dos alunos receptores. Desta forma os alunos que estavam envolvidos em um relacionamento que os supriam de material para o plágio serão motivados a procurar novos alunos ou conforme explicitado por Daly (2005) deverão confrontar suas deficiências. É esperado que a procura por novos fornecedores naturalmente promova a interação entre os alunos e crie oportunidades para a aprendizagem cooperativa. Neste modelo a busca por novos fornecedores é auto-regulada pelo estudante.

A cada nova entrega de artefatos feitos pelos alunos ao professor, o procedimento irá computar as novas intervenções a serem realizadas e irá registrar a evolução das ações de plágio entre os alunos. O registro irá demonstrar a evolução dos relacionamentos, o sucesso do método para inibir o plágio e principalmente alertará preventivamente sobre os alunos que necessitam de uma intervenção direta do professor. Como o sistema neutraliza a fonte de recurso de plágio a cada rodada de apresentação das atividades, é esperado que o aluno encontre cada vez mais resistência entre os seus colegas para plagiar. Contudo como o sistema minimiza o número de intervenções a cada rodada de atividades é esperado que exista um número suficiente de opções para que o aluno possa encontrar um colega disposto a compartilhar informações durante a maior parte da

disciplina. Desta forma o procedimento deve garantir que a dinâmica da classe não convirja para uma situação onde todos os alunos possuem atividades individuais, já que atividades comuns são reconhecidas neste trabalho como um veículo para estabelecer uma comunicação. Além disso, atividades diferentes para cada aluno é um problema conhecido que sobrecarrega o professor e inviabiliza o modelo de ensino em massa.

A proposta apresentada também tem o mérito de reconhecer a natureza social do plágio e aplicar este conceito ao problema do ensino de programação. Repensar o problema do plágio pelo prisma social é uma oportunidade para os professores agirem preventivamente no resgate dos alunos, na formulação de ambientes e atividades mais adequadas ao aprendizado, além de incentivarem os docentes a refletirem sobre as oportunidades que estão contidas na linha tênue que relaciona o plágio e a colaboração em sala de aula.

1.2 HIPÓTESES

Esta pesquisa dedica-se a estudar situações de plágio onde os exercícios ou avaliações de programação são fechados no universo da classe, ou seja:

- a) Um exercício ou avaliação é desenhado especificamente para um grupo de alunos;
- b) Não existe uma solução que possa ser copiada diretamente de um repositório externo;
- c) Cada membro deve apresentar soluções individuais para as atividades propostas.

Neste contexto a hipótese H1 deste trabalho formula que existe uma correlação entre os casos de plágio de códigos fontes e a rede social formada pelos alunos em sala de aula.

A hipótese H2 formula que em decorrência da hipótese H1 é possível modelar a rede social indiretamente através da apropriação dos resultados produzidos pelas ferramentas de detecção de plágio nos códigos fontes. Desta forma a similaridade apontada por tais ferramentas entre os artefatos dos alunos representaria um vínculo social.

A hipótese H3 estabelece que seja possível utilizar a rede social produzida pela hipótese H2 para determinar intervenções que favorecerão a interação e promoverão a aprendizagem dos alunos envolvidos com disciplinas de programação. Através da aplicação de algoritmos de imunização será determinado o número mínimo de alunos que devem receber intervenções para que os outros alunos sejam motivados a criar novas conexões sociais. Estas novas conexões sociais irão ampliar a aprendizagem do aluno e irão inibir ao longo da disciplina o plágio nas atividades práticas de programação.

1.3 ORGANIZAÇÃO

Este capítulo introdutório teve o objetivo contextualizar o problema, a motivação e indicar as hipóteses desta pesquisa.

O capítulo 2 explica a importância das atividades práticas para a disciplina de programação e descreve o plágio como um fator negativo no processo de aprendizagem. Este capítulo dedica-se também a verificar os métodos de combate ao plágio de programação existente e discorre sobre a relação do plágio com os sistemas sociais e a aprendizagem.

O capítulo 3 discorre sobre o experimento realizado e apresentada a metodologia utilizada para o desenvolvimento e verificação das hipóteses. O capítulo 3 também analisa e contextualiza os resultados obtidos.

O trabalho é finalizado no capítulo 4 onde a conclusão e os trabalhos futuros são apresentados.

2. REFERENCIAL TEÓRICO

2.1 A RELEVÂNCIA DA HABILIDADE DE PROGRAMAR

A habilidade de programar efetivamente continua sendo percebida como uma competência central na formação em computação e áreas associadas (BILLS, 2007). Esta centralidade é reforçada pela importância que a competência de programar tem nos currículos de referência da Associação Brasileira de Computação e de importantes comunidades internacionais como a ACM, IEEE e AIS.

No currículo de referência da ACM, IEEE e AIS as disciplinas de programação são explicitamente declaradas uma das bases que devem sustentar todos os bons programas de bacharelado em computação e áreas afins (SHACKELFORD et. al., 2005). Estas comunidades esperam que todos os bacharéis, independente da especialidade, demonstrem entendimento dos seguintes conceitos e habilidades fundamentais relacionadas com a programação:

- a. compreender a importância dos algoritmos e das estruturas de dados;
- b. entender, conceituar e inter-relacionar o hardware de computação através da perspectiva do software.
- c. possuir as habilidades fundamentais que permitam a implementação de algoritmos e estrutura de dados em um software;
- d. possuir as competências necessárias requeridas no projeto e implementação de grande unidades estruturais que utilizem algoritmos e estruturas de dados, além das interfaces pelos quais tais unidades comunicam-se.
- e. conhecer princípios de engenharia e tecnologias que garantam que os softwares implementados são robustos, confiáveis e apropriados ao público que irá fazer uso da solução;

Ao contrário do senso comum que percebe a programação como limitada a atividade de codificação, a adequada formação em programação permite aos bacharéis serem capazes de solucionar problemas por meio de dispositivos computacionais (DENNING, 2005). Neste contexto programar é a habilidade de

conceber um algoritmo que solucione um problema dado e planejar quais recursos e representações computacionais são necessários para que tal algoritmo seja implementado, testado, refinado e mantido. A habilidade de programar também se destaca como uma atividade importante para aprimorar o pensamento abstrato e algorítmico (LU, 2009), competências que serão utilizadas pelos alunos ao longo de toda a sua graduação e vida profissional.

2.2 O ENSINO E A APRENDIZAGEM DE PROGRAMAÇÃO

Apesar da relevância da habilidade de programar, esta disciplina é historicamente percebida pelos alunos e professores como difícil de ser ensinada e aprendida (CASPERSEN, 2009). Esta percepção é reforçada na literatura através de números que sugerem que a disciplina de programação possui um nível anormal de reprovação e desistência (BENNEDSEN, 2007). Além disto, a disciplina de programação é apontada também como um dos fatores responsáveis pela evasão e redução dos egressos em cursos de computação em todo o mundo (DENNING, 2008; YARDI, 2007; MCGETTRICK et. al., 2005).

Autores sugerem que em geral os alunos não aprendem a programar dentro de níveis aceitáveis e que este problema é internacional e reconhecido em várias instituições de ensino ao redor do mundo (KÁTAIA, 2008). Por este motivo muitas pesquisas são conduzidas com o objetivo de compreender e aperfeiçoar os processos envolvidos no ensino-aprendizagem de programação. Estas pesquisas procuram desenvolver melhores currículos, métodos pedagógicos e materiais instrucionais que facilitem o ensino e atuem positivamente na aprendizagem, retenção do conhecimento e motivação dos alunos.

Do ponto de vista pedagógico os estudos existentes ainda não possuem a natureza explicativa necessária para demonstrar e prever com exatidão como os alunos aprendem a programar ou porque os métodos propostos de ensino, ou materiais instrucionais, produzem os resultados desejados (SHEARD, 2009; LISTER, 2005). Porém muitos trabalhos de pesquisa mostram como determinados ambientes, ferramentas, procedimentos e atitudes atuam positivamente na

aprendizagem do aluno (PEARS, 2007). Os resultados destas pesquisas se traduzem em importantes ferramentas de apoio ao professor, exemplo: as destinadas à avaliação automática dos programas, ambientes de programação didáticos, monitoração automática da evolução do aluno e seu estilo de aprendizagem, propostas de linguagens de programação para aprendizes, visualização e animação de algoritmos, ambientes de programação integrados em micro-mundos, sistemas tutores inteligentes e os sistemas de detecção automática de plágio.

Entre as ferramentas listadas as destinadas à detecção de plágio recebem atenção específica dos pesquisadores e docentes (MAURER, KAPPE E ZAKA, 2006). Em razão da relação entre necessidade da prática de programação com o desenvolvimento da competência de programar, as ferramentas de detecção de plágio são uma tentativa da academia de minimizar o impacto negativo e a ameaça desta ação na aprendizagem dos alunos.

Segundo Cosma (2008) é considerado plágio em atividades práticas de programação:

- a. Apresentar ou copiar integralmente, parcialmente ou com modificações os códigos de programas e comentários técnicos produzidos por terceiros e alegar autoria própria.
- b. Traduzir o código de uma plataforma para outra mantendo parte ou totalidade dos elementos do original sem a devida citação da origem do código.
- c. Utilizar um projeto de interface gráfica com o usuário que já tenha sido concebido por outro sem a devida identificação.
- d. Fazer uso de entrada de dados concebido por um terceiro quando o aluno está sendo avaliado pela estratégia de teste que foi utilizada no desenvolvimento do seu programa.
- e. Reutilizar e adaptar componentes, módulos ou bibliotecas de função disponibilizados por terceiros sem a devida menção.
- f. Utilizar geradores de código quando a atividade foi planejada para ser desenvolvida pelo aluno individualmente e sem tal recurso.
- g. Utilizar um produto de software desenvolvido pelo próprio aluno em

atividades anteriores sem a devida referência (auto-plágio).

2.3 O PLÁGIO DE CÓDIGO FONTE

Estudos demonstram que o plágio é um evento recorrente entre os estudantes e que pode atingir níveis significativos tanto de frequência como de abrangência. No trabalho de Sheard et. al.(2002), o autor declara que 90% dos alunos admitiram praticar o plágio em exercícios de programação. Outros estudos sugerem que entre as atividades entregues ao professor cerca de 10 a 30 por cento contêm plágio freqüentemente (ZOBEL e HAMILTON, 2002; CULWIN, MACLEOD e LANCASTER, 2001; WAGNER, 2000).

A literatura registra vários motivos pelos quais os alunos envolvem-se ou realizam estas práticas (SHEARD ET. AL., 2002; SHEARD, CARBONE e DICK, 2003; WAGNER, 2000; ZOBEL e HAMILTON, 2002; DEVLIN, 2007), entre as recorrentes pode-se citar:

- a. A falta de tempo para completar as atividades propostas.
- b. O medo de falhar ou o custo de falhar em termos financeiros, de tempo e emocionais.
- c. A existência de pressão, valores ou expectativa do grupo.
- d. Uma tentativa de auxiliar um amigo ou colega.
- e. Uma percepção negativa em relação ao professor
- f. A falta de competência do professor na condução do processo de ensino-aprendizagem em sala de aula.
- g. As atividades propostas pelo professor estão além do conhecimento atual do aluno.
- h. O aluno não percebe o mérito da atividade proposta e não consegue entender sua relevância.
- i. Os recursos de hardware, software, biblioteca ou acesso aos professores são inadequados.
- j. A percepção da inexistência de punição.
- k. Uma tentativa explícita do aluno em atingir um resultado superior com o

emprego do menor esforço possível.

- I. Falta de regras claras e de definições precisas do que é o plágio e quais são os comportamentos permitidos, desejados e inadequados.

Uma vez que o aluno esteja motivado a incorrer em plágio este precisará encontrar uma fonte capaz de fornecer as informações necessárias. Conforme observado por Naude (2006), as seguintes fontes são recursos importantes quando o aluno pretende plagiar:

- a. Outros alunos da disciplina que estão envolvidos com o mesmo tipo de tarefa e cuja proximidade de relacionamento permita obter uma cópia da atividade.
- b. O acesso a terceiros (família, colegas, amigos ou serviços profissionais) que são capazes e estão dispostos a realizar parte ou totalidade da atividade do aluno.
- c. Através de grupos, fóruns ou sites localizados na internet.
- d. O trabalho em questão já foi resolvido em anos anteriores e pode ser acessado através de algum registro da própria instituição ou por meio dos alunos que já cursaram aquela disciplina.
- e. Através da apropriação indevida ou furto do trabalho de outro aluno.

Após ter decidido positivamente pelo plágio, geralmente o aluno tentará ocultar no material entregue ao professor as referências a este comportamento. A forma mais simples empregada pelos alunos para ocultar o plágio em programas é a mudança da formatação do código fonte. A inclusão ou a remoção de espaços em branco e comentários enquadra-se nesta categoria. Outras mudanças possíveis são ilustradas através do exemplo apresentado por Liu et. al., (2006) na figura 1.

<pre> 01 static void 02 make_blank (struct line *blank, int count) 03 { 04 int i; 05 unsigned char *buffer; 06 struct field *fields; 07 blank->nfields = count; 08 blank->buf.size = blank->buf.length = count + 1; 09 blank->buf.buffer = (char*) xmalloc (blank- >buf.size); 10 buffer = (unsigned char *) blank->buf.buffer; 11 blank->fields = fields = (struct field *) xmalloc (sizeof (struct field) * count); 12 for (i = 0; i < count; i++){ 13 ... 14 } 15 } </pre>	<pre> 01 static void 02 fill_content(int num, struct line* fill) 03 { 04 (*fill).store.size = fill->store.length = num + 1; 05 struct field *tabs; 06 (*fill).fields = tabs = (struct field *) xmalloc (sizeof (struct field) * num); 07 (*fill).store.buffer = (char*) xmalloc (fill- >store.size); 08 (*fill).ntabs = num; 09 unsigned char *pb; 10 pb =(unsigned char *) (*fill).store.buffer; 11 int idx = 0; 12 while(idx < num){ // fill in the storage 13 ... 14 for(int j = 0; j < idx; j++) 15 ... 16 idx++; 17 } 18 } </pre>
---	--

Figura 1 - Programa original à esquerda e programa alterado a direita (LIU et. al, 2006)

- Alteração de identificadores:* nome de variáveis, funções, procedimentos e estrutura de dados podem ser alterados facilmente sem que o programa deixe de funcionar corretamente. No exemplo acima pode-se perceber que o nome do procedimento foi alterado do original “make_blank” para “fill_content” e que a variável “fields” foi alterada para “tabs”.
- Reordenação das instruções:* algumas instruções não são dependentes e podem ser reordenadas no código mantendo o funcionamento do programa inalterado. No exemplo acima a linha 11 do código original foi realocada para a linha 6 do novo programa. Em contraste as linhas de 8 até 10 no programa original não podem ter sua ordem relativa alterada em função da sua dependência. No novo programa estas linhas mantêm sua ordem relativa e são apresentadas nas posições 4, 7 e 10.
- Substituição por instrução equivalente:* um comando de iteração do tipo “for” pode ser substituído pelo seu equivalente “while”. No exemplo acima

esta alteração pode ser percebida. No programa original a seqüência do comando “for” que começa na linha 12 foi substituído na nova versão por um “while” equivalente. Da mesma forma as expressões lógicas que controlam a instrução de decisão “if” pode ser reescritas por expressões equivalente. Por exemplo, uma seqüência `if (a){ A } else { B }` pode ser reescrita através da sua negação `if (!a) { B } else { A }`.

- d. *Inserção de código redundante ou inócuo*: um comando pode ser inserido no programa simplesmente para confundir o avaliador, já que uma seqüência de instruções que não alteram o comportamento original do sistema pode ser facilmente implementada. No exemplo acima uma instrução “for” foi inserida na linha 14 e não possui qualquer utilidade para o fluxo do programa.

Apesar das táticas apresentadas para ocultar o plágio no código fonte possam ser descobertas por meio da análise do professor, a detecção manual do plágio somente é viável em pequenos grupos de alunos (FREIRE, 2008).

Por exemplo, na hipótese de que o plágio não possa ser realizado de fonte externa e limita-se a própria classe de alunos o processo de detecção requer a comparação de todos os trabalhos entre si. Em uma turma de 50 alunos o número de comparações necessárias para garantir que todas as combinações possíveis de arquivos sejam verificadas ultrapassa 1000 comparações. Obviamente o processo de comparação citado possui custo $O(n^2)$, ou seja, sendo n o número total de trabalhos a quantidade de comparações é dado por:

$$\frac{n(n-1)}{2}$$

Por meio deste exemplo é possível perceber porque as técnicas de detecção automática de plágio são percebidas pela comunidade de computação como uma ferramenta útil para o professor. As ferramentas de detecção não fazem julgamentos, o que é responsabilidade do professor, mas procuraram apontar similaridades suspeitas e minimizar o esforço total necessário à detecção de plágio.

Muitos trabalhos sobre ferramentas capazes de detectar plágio em código

fonte já foram propostas. Entre as mais conhecidas destacam-se os trabalhos: MOSS (BOWYER, 1999), SIM (GITCHELL, 1998), JPLAG (PRECHELT, 2002) e YAP (WISE, 1996). Estes trabalhos enfocam a questão do plágio principalmente acadêmico.

É importante destacar que apesar do rótulo sobre plágio atribuído a estas ferramentas, estas mantêm correlação direta com outra área de pesquisa conhecida como detecção de clones. Estas pesquisas cuja origem é o departamento de engenharia de software procuram identificar ao longo do ciclo de vida dos produtos de software o surgimento de códigos similares. A principal função deste tipo de detecção para a engenharia é minimizar o esforço de manutenção do software e apontar áreas que poderiam beneficiar-se da criação de componentes. Apesar da motivação do surgimento de tais códigos serem distinta entre os grupos de produção de software e o dos indivíduos engajados em ações de plágio, ambas as técnicas procuram similaridades no código fonte e são correlatas (BURD, 2002).

2.4 TECNOLOGIAS DE DETECÇÃO DE PLÁGIO

Os algoritmos e técnicas de detecção de plágio podem ser classificados em dois grandes grupos (STEIN, 2007) de acordo com a ênfase:

- a) Ênfase no corpus: O artefato é comparado contra outros artefatos de um repositório conhecido através de uma função de similaridade. O repositório pode ser interno ou externo a organização. Por exemplo, o trabalho de cada aluno é comparado contra todos os outros documentos produzidos pela classe; os fragmentos ou a totalidade do trabalho de cada aluno são submetidos a uma máquina de busca na Internet, pesquisados em um repositório de programas “open-source” ou em uma base de dados mantida pela própria instituição ou fornecedor externo.
- b) Ênfase no estilo: A forma como cada pessoa apropria-se da linguagem é única, própria do indivíduo. Essa característica pode ser utilizada para identificar o plágio do artefato de duas formas principais: identificando mudanças de estilo suspeitas ao longo do próprio artefato isoladamente;

ou por meio de amostras previamente catalogadas sobre o autor é possível determinar se existe uma relação de autoria em outros artefatos anônimos.

Dentro do grupo com ênfase no corpus a literatura registra várias técnicas para determinar a similaridade entre programas. Estas técnicas são subdivididas em cinco subgrupos (ROY, 2009; LIU et. al., 2006): texto, léxico, árvores sintáticas, métricas e grafos. Por exemplo, as ferramentas JPLAG e MOSS, as mais citadas sobre detecção de plágio em programação, são representantes da classe de ferramentas léxicas.

Já o grupo com ênfase no estilo segue um processo em duas etapas: extração de características e verificação da autoria (FRANTZESKOU, 2004). Na fase de extração de características vários atributos podem ser utilizadas para criar um perfil do autor do programa, entre estas destacam-se: o layout, o estilo e a estrutura. Na etapa de verificação o perfil produzido é comparado contra os artefatos que necessitam de determinação de autoria através de técnicas estatísticas, de aprendizagem de máquina ou através de propriedades matemáticas relacionados ao espaço vetorial.

2.4.1 Detecção por comparação textual

Esta técnica utiliza-se de pouco ou nenhum pré-processamento do código fonte para a detecção de plágio. Cada linha de código é percebida como uma seqüência de caracteres, ou seja, não existe uma interpretação da linguagem na qual o programa foi escrito (do inglês parse). Uma vez que os espaços em branco sejam eliminados do arquivo, as linhas de código de todos os programas são examinadas em busca de seqüência de caracteres coincidentes.

O processo de identificação das seqüências semelhantes é orientado por técnicas padrões de busca de cadeias de caracteres, como por exemplo: o algoritmo “Longest Common Subsequence” (ROY, 2008). Esta comparação então pode ser representada através do diagrama bidimensional (Figura 2) que auxilia a visualização de semelhanças (KOSCHKE, 2008). Por meio destes diagramas é possível identificar os seguintes padrões:

- a. Diagonais completas são copias exatas;

- b. Diagonais onde itens intermediários da linha não aparecem representam mudanças;
- c. Deslocamento da diagonal para baixo ou para cima indica inserção ou deleção de código;

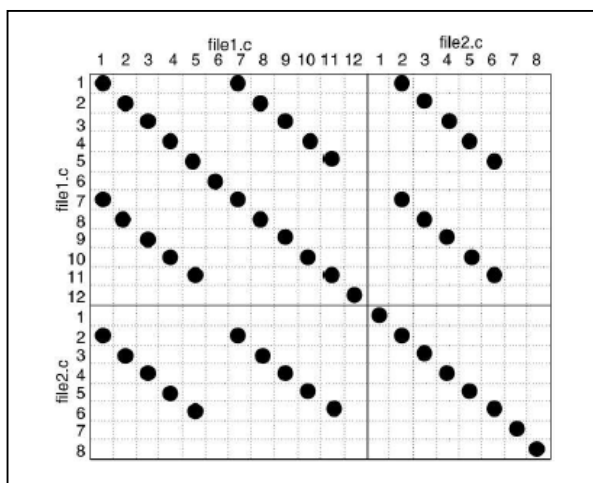


Figura 2 - Visualização de similaridades (KOSCHKE, 2008)

As técnicas baseadas somente no texto possuem duas vantagens importantes: não precisam de conhecimento léxico, sintático ou semântico da linguagem de programação que está sendo analisada e são robustas contra as iniciativas de plágio que utilizam mudanças de formatação. Contudo este tipo de técnica não é capaz de identificar as seguintes táticas para ocultar o plágio (LIU et. AL., 2006): alteração de identificadores, reordenação das instruções, substituição por instrução equivalente e inserção de código redundante ou inócuo.

2.4.2 Detecção por comparação léxica

Nesta classe de técnicas a comparação não é feita diretamente utilizando as seqüências de caracteres que compõem o código. Antes da comparação, todo o código fonte é submetido a um processo de interpretação que utiliza o conhecimento léxico da linguagem em questão. Assim cada seqüência de caracteres é substituída por sua classificação léxica correspondente na linguagem.

Após esse pré-processamento, um algoritmo de busca de seqüências de caracteres semelhantes é aplicado sobre a versão léxica do programa (Figura 3).

Por exemplo, no sistema JPLAG (PRECHELT, 2002) de detecção de plágio é utilizado o algoritmo “Greedy String Tiling” desenvolvido por Wise (1993) para determinar seqüência semelhante de caracteres.

1	public class Count {	BEGINCLASS
2	public static void main(String[] args)	VARDEF,BEGINMETHOD
3	throws java.io.IOException {	
4	int count = 0;	VARDEF,ASSIGN
5		
6	while (System.in.read() != -1)	APPLY,BEGINWHILE
7	count++;	ASSIGN,ENDWHILE
8	System.out.println(count+" chars.");	APPLY
9	}	ENDMETHOD
10	}	ENDCLASS

Figura 3 - Abstração Léxica do Código Java (PRECHELT, 2002)

As técnicas com ênfase no pré-processamento léxico são dependentes de linguagem, ao contrário das iniciativas puramente textuais. Essa capacidade de compreender o léxico da linguagem e agrupar várias seqüências de caracteres diferentes na mesma classificação torna este tipo de solução imune à alteração de identificadores e logicamente a mudanças de formatação. O exemplo da figura 4 mostra esta propriedade. Os vários identificadores são traduzidos na sua correspondência léxica de forma que uma tentativa de renomear as variáveis é facilmente detectada.

<pre>if (x < (n + y)) m = (y + c) - x; else m = y;</pre>		<pre>if (id < (id + id)) id = (id + id) - id; else id = id;</pre>
---	--	--

Figura 4 - Abstração Léxica dos Identificadores (PRECHELT, 2002)

Contudo devido à natureza da sua análise seqüencial estas técnicas continuam frágeis em sua capacidade de identificar as seguintes táticas para ocultar o plágio (LIU et. al., 2006): reordenação das instruções, substituição por instrução equivalente. No caso específico de inserção de código redundante ou inócua, está técnica não é capaz de identificar esta tática.

2.4.3 Detecção por Árvores Sintáticas

Esta classe de técnicas de detecção de plágio vão além da análise léxica e incluem informações sobre a análise sintática da linguagem.

O resultado da aplicação das regras sintáticas da linguagem resulta em uma estrutura de dados em árvore que é capaz de representar a hierarquia e a sequência das instruções (figura 5).

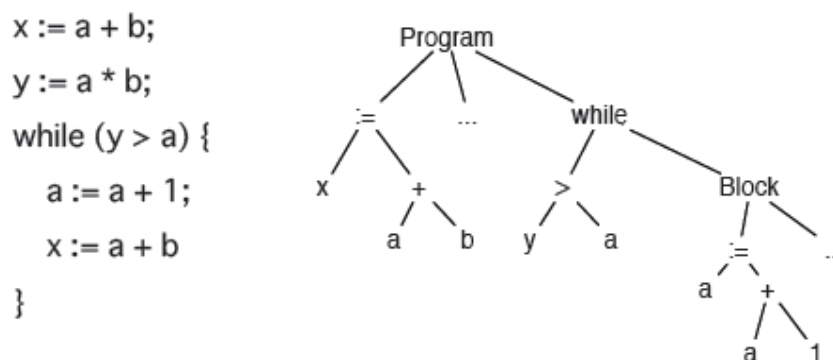


Figura 5 - Abstração Sintática de um Procedimento. À esquerda o código a direita sua representação sintática

Uma vez que o código seja representado através de sua abstração em forma de árvore sintática, algoritmos de procura de sub-árvores semelhantes percorrem todo os códigos em busca de similaridades.

Segundo Liu (2006) e levando em consideração os resultados apresentados por Roy (2009), as técnicas de detecção utilizando árvores sintáticas não são capazes de suplantar as técnicas com base em reconhecedores léxicos no momento. Além de adicionarem complexidade computacional superior (MOZGOVOY, 2006), tais ferramentas não são capazes de lidar com as seguintes táticas para ocultar o plágio: reordenação das instruções, substituição por instrução equivalente e inserção de código redundante ou inócua.

2.4.4 Detecção por Métricas

Nesta classe de ferramentas os códigos que estão sendo inspecionados não são comparados diretamente. Antes da comparação cada artefato é caracterizado por um conjunto de atributos que são quantificados. Estes atributos podem ser definidos em função de várias técnicas, como por exemplo:

- a) Métricas de engenharia de software: alguns dos atributos que podem ser utilizados incluem o número de métodos públicos, o número de métodos privados, a extensão da herança, o número de chamadas para um método, a complexidade ciclomática de McCabe, o número de parâmetros de uma função, o número de variáveis locais e o uso de várias não locais no escopo local (MANN, 2006; PATENAUDE, 1999).
- b) Atributos estáticos do código fonte: entre estes destacam-se os espaços em branco e tabulações, a diferença no número de linhas de cada artefato, as palavras com erros ortográficos e o número de palavras similares (ENGELS, 2007).
- c) Técnicas de recuperação de informação: neste modelo cada elemento textual que descreve o artefato é utilizado como característica. Uma matriz $m \times n$ é criada a partir desta interpretação. Cada coluna desta matriz descreve um dos documentos da coleção e as linhas representam o número de ocorrências de uma determinada palavra em cada documento. A lista de todos os atributos possíveis é gerada em função da coleção que esta sendo comparada (NAKOV, 2000).
- d) Funções Hash: os textos que representam cada artefato são transformados em identificadores numéricos através de funções hash. Esses identificadores passam então a caracterizar os documentos (SCHLEIMER, 2003).

Após a quantificação e caracterização do código fonte, a próxima etapa inclui processos de medição de similaridade. Esta medição é geralmente efetuada através da aplicação de álgebra vetorial. Uma técnica frequentemente utilizada é calcular a distancia euclidiana (KONTOGIANNIS, 1997) entre os elementos para determinar sua similaridade. Para isto a equação abaixo é utilizada e p e q representam dois documentos sendo comparados:

$$\sqrt{\sum_0^i (p_i - q_i)^2}$$

Além da técnica algébrica destacam-se as técnicas de agrupamento como as que são produzidas através de redes neurais artificiais (DAVEY, 1995; ENGELS, 2007).

Geralmente as técnicas com ênfase em métricas não são aplicadas isoladamente. De uma forma geral as aplicações de métricas simples são aceitas como frágeis pela comunidade de pesquisa em plágio de código fonte. Contudo novas técnicas apresentam-se promissoras e requerem um estudo comparativo mais aprofundado (MOZGOVOY, 2006). Geralmente as técnicas de métricas são combinadas as outras técnicas descritas neste trabalho. Neste caso a avaliação feita por Roy (2009) demonstra que estas seriam capazes de identificar alterações na formatação e alteração de identificadores, contudo teria dificuldades em tratar os casos de reordenação das instruções, substituição por instrução equivalente e inserção de código redundante ou inócuo.

2.4.4.1 Detecção por Métricas LSI

A técnica conhecida com LSI (do inglês Latent Semantic Index) é uma técnica utilizada em recuperação de informação que modela o espaço de busca por documentos ou passagens similares utilizando-se de conceitos da álgebra linear (DEERWESTER, 1990). Alguns trabalhos anteriores utilizaram esta técnica como indicador de plágio de código fonte e de textos (NAKOV, 2000) e mais recentemente na pesquisa de detecção de clones em software (GRANT, 2009).

Esta técnica representa cada documento de uma coleção através dos seus termos ou palavras constituintes. Este processo de caracterização cria uma matriz esparsa A onde as linhas m representam os termos ou palavras encontrados na coleção e as colunas n cada documento analisado, ou seja, cada elemento a_{ij} da matriz representa o número de vezes que o termo ou palavra i aparece no documento j :

$$A_{ij} = \begin{pmatrix} a_{11} & a_{12} & a_{13} & a_{1n} \\ a_{21} & a_{22} & a_{23} & a_{2n} \\ \dots & \dots & \dots & \dots \\ a_{m1} & a_{m2} & a_{m3} & a_{mn} \end{pmatrix}$$

Os autores Grossman e Frieder (2004) mostram um exemplo desta construção para esclarecer a mecânica do procedimento utilizando os documentos hipotéticos abaixo e seus respectivos conteúdos:

- a. documento d1: Shipment of gold damaged in a fire.
- b. documento d2: Delivery of silver arrived in a silver truck.
- c. documento d3: Shipment of gold arrived in a truck.

Dado os documentos listados acima e seus respectivos conteúdos a matriz A termo por documento é construída através da contagem do número de ocorrências de cada termo i no documento j . Os termos identificados na coleção são apresentados abaixo no vetor T para acompanhamento. Neste exemplo os autores adotaram as seguintes considerações no momento de determinar a matriz A termo por documento:

- a. as palavras relacionadas a construções gramaticais de ligação ou numeração (do inglês stopwords) não foram removidas. Ex.: Os conectores of, in, a permanecem no vetor de termos.
- b. todas as palavras foram transformados em minúsculas. Ex.: O termo Delivery é transformado em delivery.
- c. as palavras não são convertidas para sua forma léxica original (do inglês stemming). Ex.: O verbo Arrived (chegar em inglês) não é transformado em Arrive (forma do verbo não flexionada).

$$T = \begin{pmatrix} damaged \\ delivery \\ fire \\ gold \\ shipment \\ arrived \\ truck \\ silver \\ in \\ a \\ of \end{pmatrix} \quad A_{ij} = \begin{pmatrix} 1.0 & 0.0 & 0.0 \\ 0.0 & 1.0 & 0.0 \\ 1.0 & 0.0 & 0.0 \\ 1.0 & 0.0 & 1.0 \\ 1.0 & 0.0 & 1.0 \\ 0.0 & 1.0 & 1.0 \\ 0.0 & 1.0 & 1.0 \\ 0.0 & 2.0 & 0.0 \\ 1.0 & 1.0 & 1.0 \\ 1.0 & 1.0 & 1.0 \\ 1.0 & 1.0 & 1.0 \end{pmatrix}$$

Figura 6 - Exemplo de Grossman e Frieder (2004) para o procedimento LSI

Uma vez que a matriz termo por documento tenha sido formulada muitos autores sugerem aplicar um fator de correção ao valor absoluto da frequência para distribuir adequadamente a importância de determinados termos em comparação com outros da coleção. Desta forma é possível reduzir a importância de termos que aparecem frequentemente e enfatizar os termos mais únicos em cada arquivo.

Várias técnicas para calcular os fatores de correção são encontradas na literatura. Estas técnicas podem atuar na coluna, ou seja, somente aplicando um fator ao longo do documento (ajuste local), na linha onde ajusta a frequência do termo em relação a toda a coleção (ajuste global) ou em ambas simultaneamente. Algumas técnicas são (OPHIR e BROWNE, 2005):

Técnicas de Ajuste Local

Binária $a_{ij} = 1$ se o termo existe no documento, caso contrário será atribuído 0

Frequência do Termo: a_{ij} é o número de ocorrências do termo i no documento j .

Logarítmica: $a_{ij} = \log(tf_{ij} + 1)$ onde tf_{ij} é o número de ocorrências do termo i no documento j .

Augnorm

$$l_{ij} = \frac{\left(\frac{tf_{ij}}{\max_i(tf_{ij})}\right) + 1}{2}$$

Técnicas de Ajuste Global

Entropia

$$g_i = 1 + \sum_j \frac{p_{ij} \log p_{ij}}{\log n}, \text{ where } p_{ij} = \frac{tf_{ij}}{gf_i}$$

Normal

$$g_i = \frac{1}{\sqrt{\sum_j \text{tf}_{ij}^2}}$$

Idf

$$g_i = 1 + \log_2 \frac{n}{\text{df}_i}$$

No exemplo hipotético dado nesta seção para ilustrar o método, a aplicação do fator de correção por entropia resultaria na nova configuração de valores para a matriz A, conforme abaixo A passa a A':

$$T = \begin{pmatrix} \text{damaged} \\ \text{delivery} \\ \text{fire} \\ \text{gold} \\ \text{shipment} \\ \text{arrived} \\ \text{truck} \\ \text{silver} \\ \text{in} \\ \text{a} \\ \text{of} \end{pmatrix} \quad A_{ij} = \begin{pmatrix} 1.0 & 0.0 & 0.0 \\ 0.0 & 1.0 & 0.0 \\ 1.0 & 0.0 & 0.0 \\ 1.0 & 0.0 & 1.0 \\ 1.0 & 0.0 & 1.0 \\ 0.0 & 1.0 & 1.0 \\ 0.0 & 1.0 & 1.0 \\ 0.0 & 2.0 & 0.0 \\ 1.0 & 1.0 & 1.0 \\ 1.0 & 1.0 & 1.0 \\ 1.0 & 1.0 & 1.0 \end{pmatrix} \rightarrow A'_{ij} = \begin{pmatrix} 0,693 & 0,000 & 0,000 \\ 0,000 & 0,693 & 0,000 \\ 0,693 & 0,000 & 0,000 \\ 0,255 & 0,000 & 0,255 \\ 0,255 & 0,000 & 0,255 \\ 0,000 & 0,255 & 0,255 \\ 0,000 & 0,255 & 0,255 \\ 0,000 & 0,109 & 0,000 \\ 0,000 & 0,000 & 0,000 \\ 0,000 & 0,000 & 0,000 \\ 0,000 & 0,000 & 0,000 \end{pmatrix}$$

Figura 7 - Exemplo de Grossman e Frieder (2004) para o procedimento LSI com fator de correção

Após a matriz termo documento ser recalculada utilizando um fator de ponderação à técnica LSI propõe a decomposição da matriz A através do procedimento SVD (do inglês Singular Value Decomposition). O procedimento SVD é uma técnica padrão para a decomposição de matrizes em álgebra linear e foi introduzido por Golub e Kahan (1965) para calcular os valores singulares, pseudo-inversa e o posto de uma matriz. A equação proposta desta técnica é dada por:

$$A = U * S * V^t$$

onde:

- U é a matriz onde as colunas são os autovetores da matriz $A \cdot A^t$ (sendo que A^t é a transposta de A). A matriz U também é conhecida pelo nome de autovetores à esquerda.
- S é a matriz onde os elementos da diagonal são os valores singulares da matriz A .
- V é a matriz onde as colunas são os autovetores da matriz $A^t \cdot A$. A matriz V também é conhecido pelo nome de autovetores à direita.
- V^t é a transposta de V .

O resultado da decomposição pela técnica SVD permite o calculo de uma nova matriz A^* através da redução do número de valores singulares encontrados na matriz original A . A nova matriz A^* é formada então por somente k valores singulares em relação ao original conforme a figura 8, (KONTOSTATHIS, 2006):

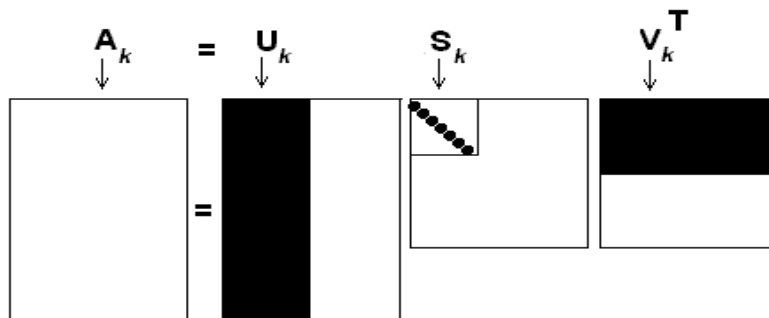


Figura 8 - Visualização gráfica de Kontostathis (2006) para o método LSI

A redução da dimensão é apresentada na literatura como um processo de redução do ruído nos dados e que permite também descobrir e enfatizar estruturas que não poderiam ser percebidas através da matriz original. A literatura também registra que não é conhecido um método para determinar k , sendo que o valor de k é arbitrário e pode ser dependente do problema modelado.

Utilizando os dados hipotéticos apresentados abaixo para exemplificar a técnica e utilizando k igual a 2 para reduzir a dimensão da matriz termo documento é possível perceber que a técnica atribui valores que não estavam inicialmente relacionados, enfatiza ou mesmo reduz a importância dos termos no documento (Figura 9).

$$A'_{ij} = \begin{pmatrix} 0,693 & 0,000 & 0,000 \\ 0,000 & 0,693 & 0,000 \\ 0,693 & 0,000 & 0,000 \\ 0,255 & 0,000 & 0,255 \\ 0,255 & 0,000 & 0,255 \\ 0,000 & 0,255 & 0,255 \\ 0,000 & 0,255 & 0,255 \\ 0,000 & 0,109 & 0,000 \\ 0,000 & 0,000 & 0,000 \\ 0,000 & 0,000 & 0,000 \\ 0,000 & 0,000 & 0,000 \end{pmatrix} \quad A^{**}_{ij} = \begin{pmatrix} 0,677 & -0,008 & 0,102 \\ -0,008 & 0,688 & 0,055 \\ 0,677 & -0,008 & 0,102 \\ 0,287 & 0,017 & 0,045 \\ 0,287 & 0,017 & 0,045 \\ 0,034 & 0,274 & 0,027 \\ 0,034 & 0,274 & 0,027 \\ -0,013 & 1,091 & 0,087 \\ 0,0 & 0,0 & 0,0 \\ 0,0 & 0,0 & 0,0 \\ 0,0 & 0,0 & 0,0 \end{pmatrix}$$

Figura 9 - Redução usando k igual a 2

Na ilustração abaixo (figura 10) há uma comparação entre a redução da matriz A e sua versão A' com o fator de correção por entropia:

$$A^{*ij} = \begin{pmatrix} 0,665 & -0,147 & 0,447 \\ -0,147 & 0,934 & 0,198 \\ 0,665 & -0,147 & 0,447 \\ 1,113 & 0,050 & 0,847 \\ 1,113 & 0,050 & 0,847 \\ 0,300 & 1,132 & 0,597 \\ 0,300 & 1,132 & 0,597 \\ -0,295 & 1,869 & 0,396 \\ 0,966 & 0,985 & 1,045 \\ 0,966 & 0,985 & 1,045 \\ 0,966 & 0,985 & 1,045 \end{pmatrix} \quad A^{**ij} = \begin{pmatrix} 0,677 & -0,008 & 0,102 \\ -0,008 & 0,688 & 0,055 \\ 0,677 & -0,008 & 0,102 \\ 0,287 & 0,017 & 0,045 \\ 0,287 & 0,017 & 0,045 \\ 0,034 & 0,274 & 0,027 \\ 0,034 & 0,274 & 0,027 \\ -0,013 & 1,091 & 0,087 \\ 0,0 & 0,0 & 0,0 \\ 0,0 & 0,0 & 0,0 \\ 0,0 & 0,0 & 0,0 \end{pmatrix}$$

Figura 10 - Comparação entre as matrizes com e sem redução

Levando em consideração que S é simétrico em relação a sua diagonal os autores Berry, Dumais e O'Brien propuseram que qualquer vetor q representando uma pesquisa ou um documento externo a coleção inicial poderia ser codificado em relação a matriz A'* da seguinte forma:

$$q^k = q^t \times U^k \times \left(\frac{1}{S^k} \right)$$

Onde: q^k é a representação de q no espaço k

q^t é transposta de q

U^k é U reduzido a k

S^k é S reduzido a k

Por exemplo, na Figura 11 dado que q representaria o vetor que codifica o seguinte documento: shipment arrived in a truck; a equação acima transformaria q em sua respectiva representação q_k na dimensão reduzida k da seguinte forma:

$$T = \begin{pmatrix} \text{damaged} \\ \text{delivery} \\ \text{fire} \\ \text{gold} \\ \text{shipment} \\ \text{arrived} \\ \text{truck} \\ \text{silver} \\ \text{in} \\ \text{a} \\ \text{of} \end{pmatrix} \quad q = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 1 \\ 0 \\ 1 \\ 1 \\ 0 \end{pmatrix} \rightarrow q_k = q_t \times U_k \times \left(\frac{1}{S_k}\right) \rightarrow q_k = (0,4152 \quad 0, -5457)$$

Figura 11 - Exemplo de Redução do vetor

Através da codificação na base adequada é possível aplicar um processo de comparação entre um vetor q e qualquer documento d da coleção. Como consequência da formulação de Berry, Dumais e O'Brien é possível estabelecer que:

$$f(q_k, d_k) = f\left(q_t \times U_k \times \left(\frac{1}{S_k}\right), d_t \times U_k \times \left(\frac{1}{S_k}\right)\right)$$

Na equação acima q_t e d_t são respectivamente a transposta de q e d . A função f é uma função para calcular a similaridade entre os dois vetores.

A função de similaridade geralmente empregada em conjunto com a técnica LSI é a medida do cosseno do ângulo formado entre os vetores dado por:

$$\cos(\alpha) = \frac{(q \cdot d)}{(\|q\| \|d\|)}$$

Onde:

$(q \cdot d)$ é o produto interno

$(\|q\| \|d\|)$ é a norma dos vetores.

2.4.5 Detecção por Grafos

Esta técnica combina as etapas léxicas e sintáticas e inclui a análise semântica do código para representar não só as estruturas de dados, mas também o fluxo do programa (HORWITZ, 1990). A partir estas informações o programa é representado através de um grafo (Figura 12). No exemplo abaixo as linhas solidas são dependências de dados e as linhas tracejadas são dependências de controle.

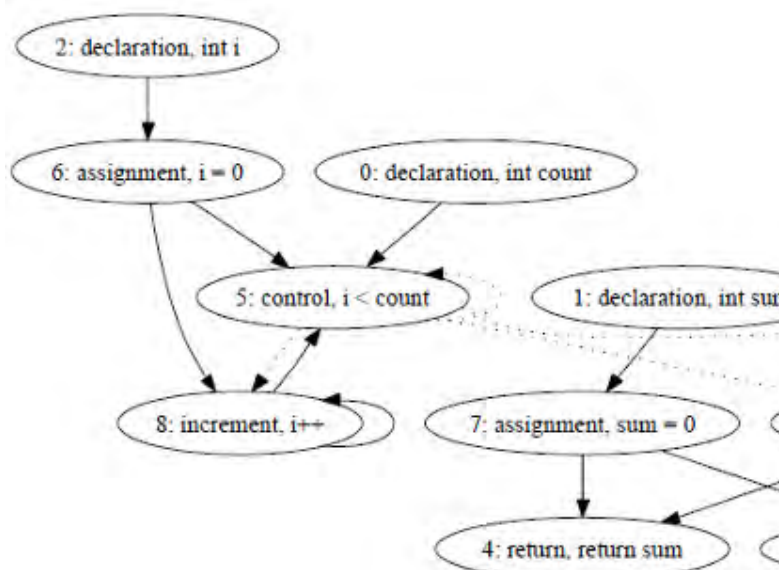


Figura 12 - Exemplo de um grafo construído a partir de um programa (LIU et. al, 2006)

A existência de sub-grafos isomórficos entre dois ou mais artefatos revelaria a existência de similaridades.

Esse procedimento mostra-se capaz de lidar com todas as táticas utilizadas para ocultar o plágio, ou seja: identificar alterações na formatação, alteração de identificadores, reordenação das instruções, substituição por instrução equivalente e inserção de código redundante ou inócuo (LIU et al. , 2006; ROY, 2009).

Recentemente Gabel et al.(2008) apresentaram um procedimento que segundo os autores apesar da complexidade envolvida na detecção de similaridades com grafos, a técnica proposta é escalável. Isto permitiria a computação de grande quantidade de dados em tempo aceitável.

2.4.6 Detecção por Estilometria

Um dos problemas apontados nas técnicas com ênfase no corpus é que elas não são capazes de lidar diretamente com as seguintes situações (OHNO, 2008):

- a. O aluno pode contratar um serviço externo, ou convencer um terceiro a realizar a atividade em seu lugar.
- b. O aluno pode usar uma fonte de informações ampla como a internet
- c. O aluno pode encontrar uma fonte externa que não está acessível publicamente, como por exemplo: os códigos da empresa onde trabalha.
- d. A existência de similaridades naturais que surgem espontaneamente em sala de aula e que são difíceis de serem diferenciadas do plágio. Tais situações acontecem devido ao tipo de atividade que é compartilhada por todos e ao contexto.

Para minimizar a situação de plágio com origem na Internet alguns autores propuseram que é fundamental o desenho das atividades e dos instrumentos de avaliação de forma a inibirem o plágio. A construção de atividades que possuem parâmetros específicos, ou combinações específicas tornam o plágio direto da Internet mais difícil. Conforme expresso por Carroll (2001), as seguintes atitudes são boas práticas para prevenir o plágio: renovar periodicamente as atividades e instrumentos de avaliação, incluir outros elementos de avaliação que possam ser complementares aos trabalhos práticos ou descritivos, criar atividades individualizadas, integrar os instrumentos de avaliação de forma que estes contenham uma estrutura de referência cruzada e providenciar várias avaliações ao longo da disciplina.

Outros autores propuseram ferramentas de verificação de plágio que são construídas utilizando extensões dos serviços de busca disponíveis na Internet como as API do Google, por exemplo (NIEZGODA, 2006). Tais ferramentas expandem a capacidade de comparação para além do grupo de alunos.

Também já foi proposto para lidar com estas questões o desenvolvimento de uma ferramenta para a construção de programas próprios para a sala de aula onde

todas as atividades do aluno são monitoradas (VAMPLEW, 2005). Nestas ferramentas o que o aluno faz é registrado detalhadamente criando indícios que permitem determinar se o esforço feito condiz com o material entregue. Diferenças entre as variáveis propostas pelos autores são indicadores de plágio. Contudo nenhuma destas soluções garante uma proteção totalmente confiável contra as táticas apontadas acima.

As ferramentas com ênfase no estilo propõem resolver esta questão em definitivo. Estas técnicas forenses procuram mensurar através do artefato características como autoria, quantidade de autores, intenção do autor ou áreas de especialidade (FRANTZESKOU, 2006).

O avanço das técnicas de aprendizagem de máquina, processamento de linguagem natural e técnicas de recuperação de informação permitiram consideráveis avanços nesta disciplina. Contudo em uma revisão recente da literatura e dos trabalhos disponíveis mostra que as técnicas ainda precisam aumentar a confiabilidade dos resultados (STAMATATOS, 2008). Segundo o autor a confiabilidade destas técnicas ainda é um fator do número de autores, do tamanho dos textos e da quantidade de amostras que são utilizados para treinar o modelo para a atividade de reconhecimento.

O trabalho de Ohno (2008) apresenta um modelo para a solução destes problemas no código fonte através de um reconhecedor construído com técnicas de cadeias de Markov ocultas (Hidden Markov Models - HMM), que segundo os autores é capaz de lidar com as características dos trabalhos acadêmicos de programação e apresenta-se como um solução final as estratégias de plágio discutida acima.

2.5 OS DETECTORES E AS REDES SOCIAIS

Atualmente os algoritmos de detecção de plágio mostram para cada artefato entregue quais são os trechos de código que são suspeitos e a provável origem da fonte. A visualização destas informações é geralmente feita pelos interessados através de uma lista ordenada de forma decrescente. Os artefatos com maior

probabilidade de plágio são apresentados primeiros. Além disto, para cada elemento da lista o sistema de detecção aponta os outros trabalhos relacionados. O exemplo abaixo (Figura 13) proveniente da ferramenta JPLAG (PRECHELT, 2002) ilustra este ponto. A primeira coluna representa o nome do artefato e as colunas seguintes a dependência em relação aos demais programas apresentados para análise.

861301	->	861196 (40.1%)	862246 (35.9%)	827825 (20.4%)
861005	->	861641 (maxSimTest) (39.8%)	861061 (25.0%)	827654 (20.2%)
862564	->	862326 (35.9%)	942909 (20.7%)	862246 (20.3%)

Figura 13 - Exemplo de relatório de detecção de plágio do JPLAG (PRECHELT, 2002)

Este modelo de apresentação foi aprimorado por Freire (2008). Neste trabalho foi reconhecida a importância de analisar os dados sobre o plágio através de uma visão integrada. O autor utilizou em sua ferramenta as bibliotecas de visualização de grafos e tornou possível mostrar a conexão entre vários arquivos simultaneamente (Figura 14).

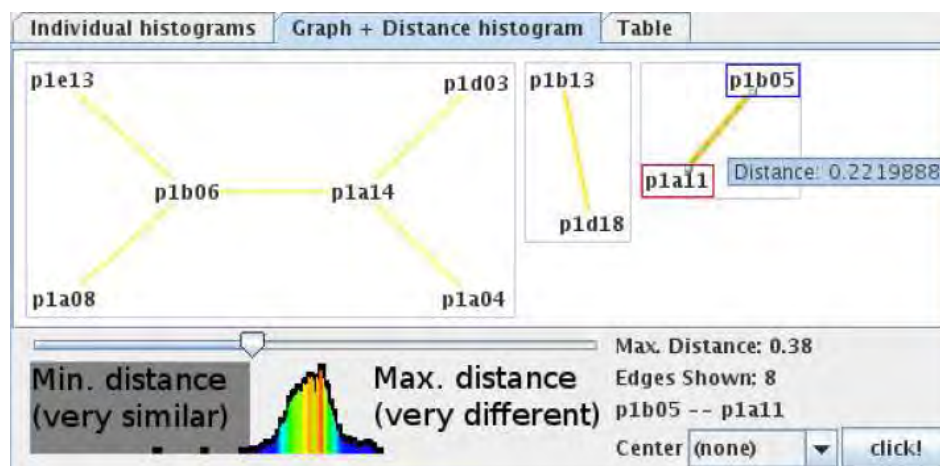


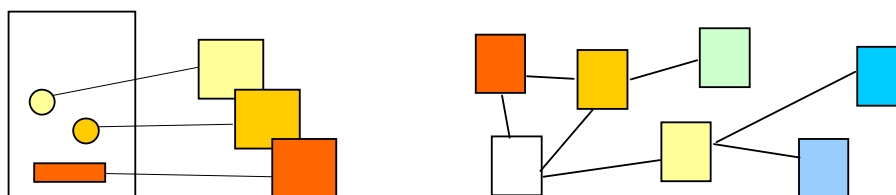
Figura 14 - Ferramenta de visualização AC (FREIRE, 2008)

Apesar do trabalho da inclusão dos grafos contribuírem com o propósito da busca por similaridades e eliminação de falsos positivos, este tipo de abordagem continua a enfatizar o plágio, os artefatos e não reconhece o comportamento social envolvido no processo.

Em acordo com observações exploratórias iniciais em sala de aula, as

similaridades representadas no grafo sugerem uma riqueza de informações maior do que a simples demonstração da ocorrência de plágio em determinados artefatos (LUQUINI e OMAR, 2009). Estas ligações podem refletir algum tipo de relacionamento entre os alunos. Desta forma a ligação entre os artefatos representaria um vínculo social entre seus proprietários.

A proposição deste trabalho formula que é possível visualizar a quantidade de plágio entre os códigos como uma medida indireta do relacionamento entre os estudantes. Ao contrário de olhar cada aluno individualmente, este trabalho enfatiza que o plágio é um veículo para modelar a rede social dos alunos. Desta forma a visão atual do plágio pode ser repensada através dos conceitos de redes sociais (Figura 15).



Enfoque atual das ferramentas de detecção de plágio: O aluno cometeu plágio? Qual a fonte utilizada?

Enfoque em rede proposta por esta pesquisa: Como os alunos estão organizados no processo de plágio?

Figura 15 - Uma nova interpretação para as ocorrências de plágio no ensino de programação (LUQUINI e OMAR, 2009).

Conforme lembrado por Zobel e Hamilton (2002) não existe uma solução definitiva para a questão do plágio e é necessário empregar múltiplas perspectivas e técnicas complementares para lidar com tal fato. Este trabalho utiliza a abordagem de redes sociais como forma de acrescentar uma nova perspectiva ao estudo do plágio nas salas de aula de programação. Além deste princípio o trabalho também propõe utilizar esta rede social como base para a criação de procedimentos que permitirão ganhos de aprendizagem.

2.6 AS REDES SOCIAIS

A rede social é uma estrutura que descreve as ligações diretas e indiretas que surgem entre as pessoas ou outras entidades. A teoria de redes sociais afirma que tais estruturas afetam o comportamento dos indivíduos que dela fazem parte e determinam as possíveis dinâmicas do grupo (MILKA, 2007).

A análise da rede social emprega técnicas matemáticas derivadas da teoria dos grafos. Esta formalização possui as seguintes vantagens (JAMALI, 2006):

- a. representação compacta e sistemática da rede;
- b. aplicação direta de técnicas computacionais;
- c. a própria teoria dos grafos sugere propriedades importantes que ajudam a esclarecer os fenômenos sociais.

Apropriando desta estrutura matemática cada vértice do grafo representa um ator ou entidade envolvida na análise e cada aresta representa uma relação que vincula dois atores. Estas relações podem modelar:

- a. similaridades: atores que estão em uma mesma localização, são membros de uma mesma organização ou são caracterizados pelo mesmo atributo;
- b. relações sociais: atores que possuem parentesco, um determinado papel na organização, um vínculo afetivo ou cognitivo;
- c. atividades: como por exemplo aconselhar, ajudar, ou conversar;
- d. fluxo: como por exemplo de informações, crença e recursos.

No exemplo abaixo é apresentada uma visão gráfica de uma rede social onde os círculos representam os atores e as linhas os relacionamentos.

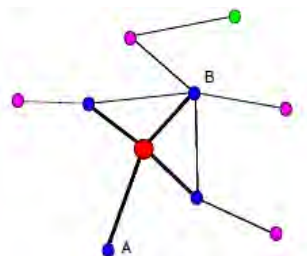


Figura 16 - Apresentação gráfica da rede social (SIH, 2009)

A visualização tem origem em uma matriz $M \times M$, onde M é cada ator que está sendo analisado. A incidência de uma relação entre é quantificada na coleta dos dados. No exemplo abaixo a existência de uma relação é pontuada pelas células com o número 1. A inexistência de relação pelo número 0;

	A	B	C	D	E	F	G
A	1	1	1	0	1	1	1
B	1	1	1	1	0	0	0
C	1	1	1	1	0	0	0
D	0	1	1	1	0	0	0
E	0	0	0	0	1	1	0
F	1	0	0	0	1	1	0
G	1	0	0	0	0	0	1

Figura 17 - Apresentação matricial da rede social (CHURCHILL,2005)

Através dos modelos, como o exemplo acima, é possível computar várias métricas que caracterizam a rede social. Entre estas métricas destacam-se as de (JAMALI, 2006):

- a. Fluxo: permite determinar a facilidade que um ator tem para enviar um recurso (material ou imaterial) para outro. Essa medida avalia a redundância e a fragilidade destas conexões e permite determinar quantitativamente um grau de ligação entre os dois atores.
- b. Sub-grupos: permite determinar os subgrupos que surgem dentro de um grupo maior e analisar como estes subgrupos afetam a rede. Perguntas importantes podem ser quantificadas como: se os sub-grupos compartilham membros; se eles fracionam a rede social; qual a dimensão dos sub-grupos; quais pessoas estão isoladas; quais atores exercem papéis importantes.
- c. Centralidade e poder: permite determinar a capacidade dos atores de influenciarem e controlarem o grupo. Apesar das definições de poder e centralidade não serem consensuais, várias medidas existem para caracterizar estas propriedades quantitativamente e entre as muitas destaca-se (Tabela 1):

Métrica	Definição	Influência
Valência ou Grau	Número de conexões que um ator possui diretamente	O ator possui mais opções e oportunidades
Centralidade	Uma medida de distância do ator até todos os outros atores da rede	Capacidade de barganha e de adquirir informações
Intermediação	Uma medida de como o ator está posicionado entre o fluxo dos demais atores	Capacidade de isolar os outros atores e controlar o fluxo de recursos

Tabela 1 - Propriedades de Centralidade e Poder (CHURCHILL,2005)

A modelagem da rede social e as métricas são capazes de caracterizar a estrutura do grupo e constituem-se em bases quantitativas para estudos correlacionais e ações de intervenção que planejam atuar sobre a dinâmica do grupo e seus resultados. Por exemplo, no trabalho de Garcia (2008) as técnicas de redes sociais são empregadas para desenhar uma solução de intervenção em equipes corporativas com a finalidade de atingir resultados gerenciais. Na pesquisa de Dawson (2008) as técnicas de redes sociais são utilizadas em um estudo para determinar a correlação entre as métricas da rede social e a percepção de inclusão dos alunos em sala de aula.

Vários métodos existem para coletar os dados para que a rede seja modelada. Entre as opções incluem-se os questionários, entrevistas, observações e fontes indiretas. Nas equipes de software a medição indireta é particularmente útil pela existência de artefatos que são produzidos continuamente pelas equipes. A engenharia de software tem desenvolvido estudos que permitem inferir a rede social através da análise dos artefatos técnicos produzidos (SARMA, 2009). Tais estudos procuram aumentar a eficiência das equipes na produção de software. Este trabalho também segue esta perspectiva e modela a rede social por meio dos detectores de plágio aplicados aos artefatos produzidos pelos alunos. Este procedimento se apóia na premissa que o plágio é uma função da relação social entre os alunos e que a

descrição de tal rede permitiria o projeto de intervenções. Estas intervenções auxiliariam na aprendizagem do aluno e na inibição dos efeitos negativos do plágio sem instigar o conflito aluno versus professor.

2.7 AS REDES SOCIAIS E A COOPERAÇÃO

A ocorrência do plágio entre os alunos é postulada neste trabalho como uma deficiência de aprendizagem e é modelada como um reflexo de uma relação cooperativa não funcional. Segundo a teoria da distração-conflito e da coesão de grupos as relações de cooperação podem privilegiar os aspectos sociais em função dos resultados. Um grupo que contenha pessoas preocupadas com o resultado atrairia mais pessoas deste tipo. Ao contrário, um grupo que privilegia as relações sociais poderia enfatizar tal comportamento em detrimento dos resultados (ROSETH, 2008). Este trabalho formula o plágio como produto de uma relação entre os alunos que premia o envolvimento social em detrimento dos resultados da aprendizagem.

Para alterar esse comportamento é necessário criar incentivos para os alunos envolverem-se com outros indivíduos em sala de aula. Para que esta procura seja natural e os alunos tenham a oportunidade de negociar seus novos relacionamentos, um procedimento inspirado nas técnicas de imunização foi desenhado (KIMURA, 2008; BORGATTI, 2006; ARULSELVAN, 2008). Este procedimento altera o conteúdo das atividades dos alunos centrais em cada sub-grupo de forma que os alunos dependentes do plágio sejam incentivados a procurarem novos relacionamentos. No exemplo acima (Figura 18) a desconexão dos nós escuros irá fragmentar a rede social e criaria um motivador para os alunos negociarem novos relacionamentos.

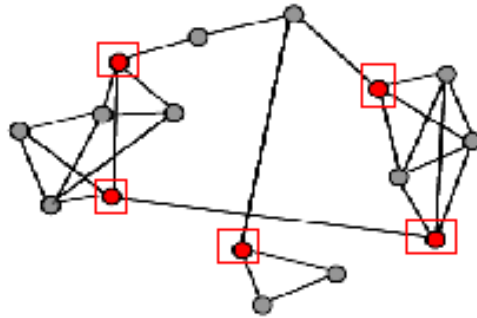


Figura 18 - Imunização dos alunos fornecedores (nós envoltos no retângulo) inspirado na metáfora de fragmentação da rede em processos de combate a epidemias

3. EXPERIMENTO

A proposta deste trabalho fundamenta-se na hipótese de que existe uma correlação entre o plágio verificado nos programas e a rede social dos alunos. Esta hipótese que foi chamada de H1 nesta pesquisa originou-se de percepções empíricas decorrentes das atividades de docência. Com o objetivo de avaliar a hipótese H1 e criar as condições para derivar as hipóteses H2 e finalmente H3, um experimento foi desenhado com o intuito de aplicar uma estrutura estatística apropriada que possa aumentar a força das colocações deste trabalho.

O experimento projetado para determinar H1 procura estabelecer que se o artefato do aluno 'a' possui uma relação de plágio com o artefato do aluno 'b', dado por $P(a,b)$, é possível encontrar na rede social S uma relação entre o aluno 'a' e 'b' de forma que a existência de $P(a,b)$ implicará em $S(a,b)$. Esta argumentação é comum em estudos de redes sociais onde procura-se determinar para um mesmo grupo de indivíduos se duas relações distintas sobre este grupo possuem correlação.

O problema de correlação é tratado em estudos de redes sociais através dos algoritmos QAP (do inglês Quadratic Assignment Procedure) (ZHANG, 2008). Neste trabalho o algoritmo QAP foi aplicado para determinar estatisticamente se as duas propriedades de interesse, ou seja, o vínculo social e uma ação de plágio estavam correlacionados dentro de uma medida estatística significativa.

3.1 A METODOLOGIA

Para quantificar o vínculo social os alunos que participaram deste experimento foram submetidos a um questionário sociométrico. O questionário sociométrico é uma ferramenta utilizada para construir a rede social em estudos envolvendo coleta direta de informações através de perguntas endereçadas ao grupo (VALENTE, 2004; TAN, 2008; MARTÍNEZ-MONÉS, 2003; KANDA, 2007). Nestas perguntas cada indivíduo da população ou da amostra registra sua opção e

sua percepção sobre os outros membros do grupo. A síntese das respostas é organizada através de um sociograma, ou seja, o equivalente a uma rede na definição matemática da teoria dos grafos (CHURCHILL, 2005).

Após o sociograma inicial ter sido construído utilizando-se somente as respostas ao questionário, cada um dos trabalhos práticos de programação dos alunos foi submetido a um algoritmo de detecção do plágio.

Finalmente a ferramenta conhecida como UCINET (HUISMAN, 2005) foi utilizada para calcular a correlação estatística através do método QAP.

3.2 A POPULAÇÃO

O experimento foi conduzido em um curso técnico em informática na região de São Paulo capital durante a disciplina de introdução a programação. Os 47 alunos que fizeram parte do experimento possuíam idades entre 18-22 anos. A duração da disciplina foi de 48 horas distribuídas ao longo de aproximadamente 03 semanas.

3.3 A DISCIPLINA

O objetivo da disciplina foi ministrar o seguinte currículo:

- a. Introduzir o processo de construção de Algoritmos;
- b. Conceituar as variáveis e constantes;
- c. Introduzir os tipo de dados elementares e matrizes;
- d. Introduzir as operações procedimentais elementares (decisão e iteração);
- e. Introduzir a decomposição de programas (funções e procedimentos);

3.4 OS ARTEFATOS

Durante a disciplina foram solicitados exercícios de programação em caráter somativo com participação de 40% na média final do aluno. Todos os trabalhos

solicitados foram entregues no ultimo dia da disciplina. Entregas intermediárias não foram aceitas. O detalhamento da natureza de cada um dos exercícios está descrito no anexo 2.



Os exercícios numerados de 1 até 9 foram feitos em laboratórios durante as aulas e os trabalhos numerados de 10 até 16 foram atribuídos como exercícios complementares que deveriam ser feitos fora das aulas regulares. Na tabela 2 está uma listagem dos exercícios, os respectivos alunos e o percentual de não entrega:

Ex.	Nome	Não Entregaram	
		Alunos	Percentual
1	Movimentar o robô da posição inicial para a posição final	14,20,26	6,38%
2	Movimentar a caixa da sua posição inicial para posição final	14,20,26	6,38%
3	Movimentar as caixas da sua posição inicial para posição final	14,20,26	6,38%
4	Faça o robô andar em círculos	14,20,26,35	8,51%
5	Faça o robô sair do labirinto	14,20,26	6,38%
6	Faça o robô mover as baterias para a posição indicada	14,20,26	6,38%
7	Faça o robô verificar a quantidade de água de cada planta	14,20,26	6,38%
8	Faça o robô calcular a média de água no grupo de plantas I	01,14,20,26,35,42	12,76%
9	Faça o robô calcular a média de água no grupo de plantas II	14,20,26	6,38%
10	Faça o robô verificar o quanto de água durante 10 segundos	05,10,14,20,25,26,42	14,89%

11	Faça o robô verificar o quanto de água durante 10 segundos II	01,05,08, 14,26,39,42	14,89%
12	Manter a planta saudável até a bateria terminar	01,02,03,05 ,06,08,10, 14,20,25,26 ,31,34,35, 39,40,42,43	38,29%
13	Vigiar a planta durante 1 minuto	10,14,26,27 ,39	10,63%
14	Vigiar as plantas durante 2 minutos	14,20,26,35 ,39	10,63%
15	Vigiar as plantas durante 2 minutos II	05,08,10,14 ,26,35,39, 47	17,02%
16	Consumir todas as baterias	08,10,14,20 ,26,31,34, 35,37,39,42 ,47	25,53%

Tabela 2 – Disposição da entrega dos trabalhos

Durante a disciplina foi aplicado um questionário sociométrico, com as perguntas listadas no anexo 3, onde os alunos apontaram com quais outros alunos desejavam trabalhar em projetos de equipe em caráter somativo. Na figura 19 é apresentado o desenho da rede social resultante deste questionário em conjunto com a nota média da prova escrita individual de cada aluno. A legenda para a rede social é descrita desta forma:

- a.  são alunos com nota inferior a 6;
- b.  são alunos com nota inferior a 7 porem maior ou igual a 6;

- c. ▲ são alunos com nota superior ou igual a 7 porem inferior a 8,5;
- d. ◆ são alunos com nota superior ou igual a 8,5.

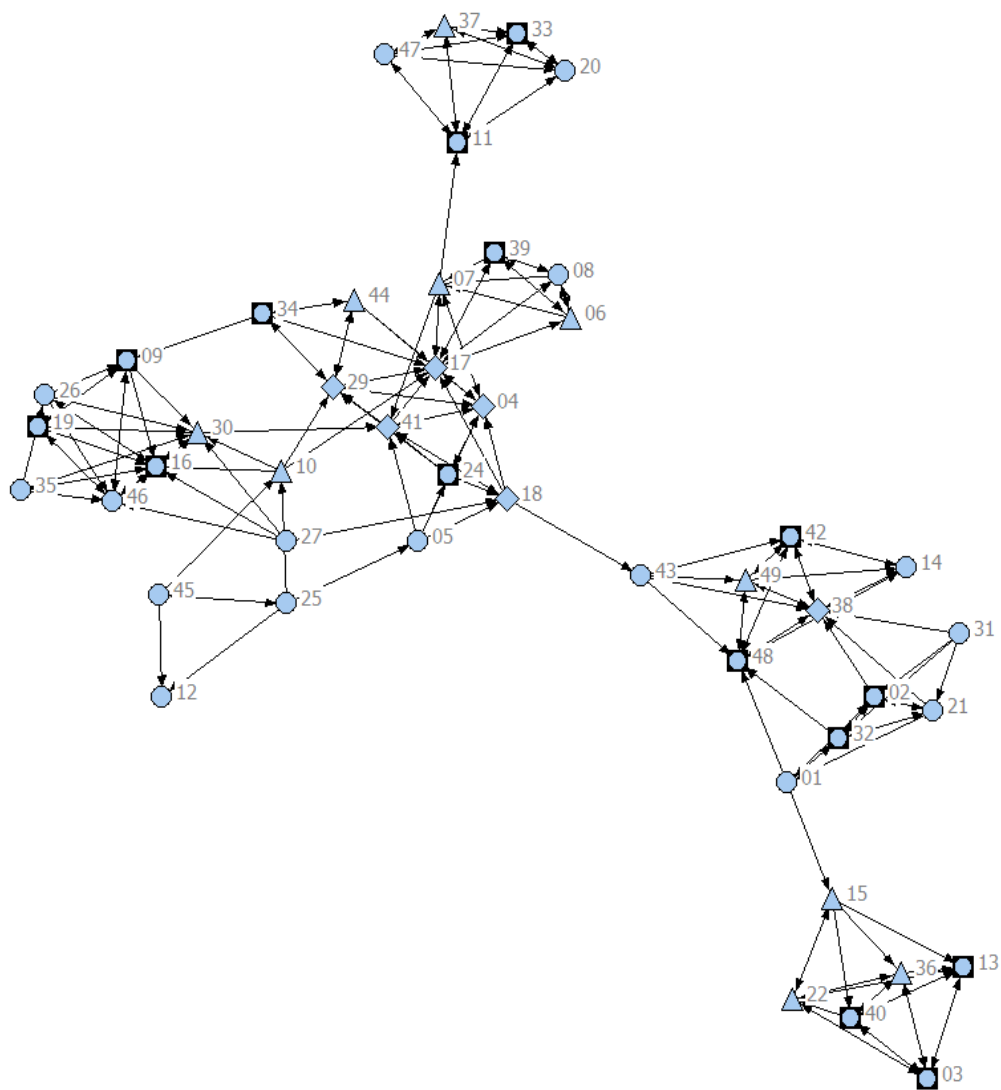


Figura 19 - Grafo sociométrico proveniente do questionário aplicado aos alunos

3.5 A FUNÇÃO PLÁGIO

Conforme exposto anteriormente a detecção do plágio exige uma ferramenta que permita apontar entre uma coleção de trabalhos quais deles possuem similaridades que não deveriam existir. Já foi apontado neste trabalho que a confirmação do plágio depende da avaliação do professor, pois existem vários motivos além do plágio para que os trabalhos de programação dos alunos possuam similaridades (Mann e Frew, 2006), entre estas se destacam:

- a. Todos os alunos aprenderam e são incentivados a seguir a mesma formatação e estilo de codificação;
- b. Os alunos estão em estágios similares de experiência e maturidade nas técnicas de programação;
- c. Os exemplos e processos para atingir-se a solução do problema que foi apresentado pelo tutor para os alunos é a mesma;
- d. Os exercícios oferecidos aos alunos são desenhados para explorar um aspecto específico do que está sendo aprendido e conseqüentemente possuem somente um número reduzido de soluções possíveis, ou mesmo uma única solução;
- e. Soluções parciais podem ter sido dadas aos alunos pelo professor como ponto de partida do trabalho ou podem ter sido explicadas ao grupo em sala de aula;
- f. O trabalho a ser feito tem como base um trabalho anterior;
- g. A linguagem de programação é por natureza um conjunto limitado da linguagem natural e por esta definição restritiva em possibilidades de representação do problema;
- h. Os alunos estão utilizando um mesmo livro texto e solucionando os problemas com base nestes;
- i. Os alunos foram direcionados a trabalharem em conjunto em partes do problema que não estão relacionadas diretamente ao programa, como a parte de planejamento e projeto da solução;
- j. O professor permitiu que o grupo dividisse o trabalho e este foi entregue individualmente com a recombinação do trabalho do grupo.

Desta forma é importante escolher uma função de plágio que seja capaz de caracterizar as similaridades e apontar aquelas que realmente são relevantes. No estudo conduzido por Mann e Frew (2006) os autores apontaram que a grande quantidade de similaridade calculada entre os trabalhos coletados não pode ser considerada como plágio, mas sim como um resultado normal da estrutura imposta pelas atividades em sala de aula e por consequência ligada as explicações acima. Os autores citam que para a verificação do plágio é importante atentar aos desvios de similaridade muito grandes em relação aos outros participantes do grupo.

Neste trabalho a função que será adotada para medir a similaridade entre os códigos é baseada na técnica LSI.

A técnica LSI como função de detecção de similaridades tem a vantagem para este trabalho de não requerer um processar léxico, sintático ou semântico e conseqüentemente poder ser utilizada em qualquer artefato produzido em sala de aula independente da linguagem de programação. Outra vantagem importante é que todos os artefatos podem ser numericamente mensurados em seu nível de similaridade. Diferentemente da mensuração binária está medida é importante, pois quantifica para cada par de artefatos seu grau de similaridade ou dissimilaridade em relação ao outro o que se esperam torne o processo de distinção de casos especiais mais apropriados.

3.6 PRÉ- PROCESSAMENTO

Para cada exercício entregue pelo aluno foi gerada a sua respectiva matriz termo documento através de uma ferramenta construída para este trabalho cujos detalhes estão no anexo 4. Esta ferramenta executa o desmembramento das palavras através de expressões regulares no padrão IEEE POSIX (1994).

O procedimento para desmembramento do código fonte estabelecido na ferramenta obedece às seguintes regras:

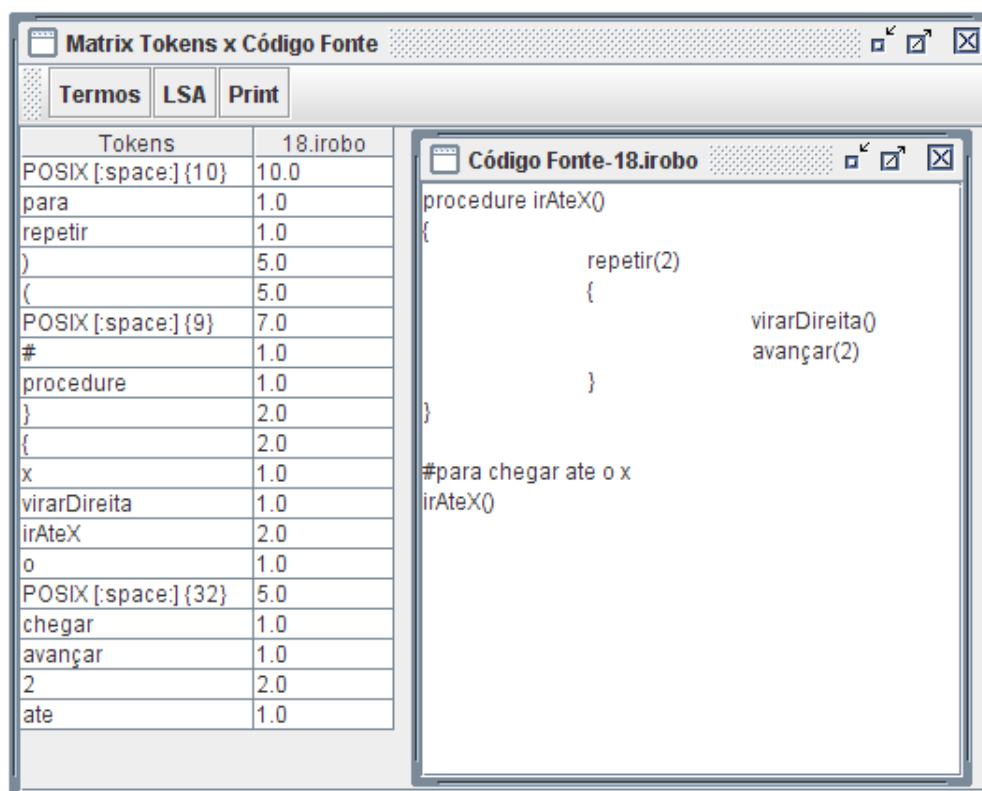
- a. Qualquer caractere que represente um numeral, não importa se este está isolado ou como parte de outra palavra, é desmembrado do código fonte e tratado na matriz termo-documento como um elemento único. O código

- IEEE POSIX para a expressão regular equivalente é `[:digit:]+`. Esta expressão significa qualquer seqüência numérica com 1 ou mais elementos;
- b. Qualquer caractere que não faça parte do conjunto definido como alfanumérico ou tabulação é tratado como caractere especial pela ferramenta e aparecerá na matriz termo documento como um elemento único. O código IEEE POSIX para a expressão regular é `[^[:alpha:][:digit:][:space:]]`. Esta expressão significa qualquer caractere desde que não seja texto, numeral ou caractere de tabulação;
 - c. Qualquer caractere representando somente texto padrão é isolado e tem uma entrada única na matriz termo documento. O código IEEE POSIX para a expressão regular é `([:alpha:])+`. Esta expressão significa qualquer seqüência de caracteres texto padrão contendo 1 ou mais ocorrências;
 - d. Com o objetivo de incorporar e enfatizar que o uso de espaços, tabulações e quebras de linha diferenciam o estilo de programação dos alunos, a matriz termo documento também terá uma entrada específica para cada elemento de tabulação definido no grupo IEEE POSIX através do código `[:space:]`. Esta expressão significa que os caracteres de tabulação presentes no código fonte serão lidos um a um e interpretados individualmente.

A regra referente às tabulações foi incorporada na caracterização da matriz termo documento dos exercícios com a finalidade de estabelecer uma oportunidade de a ferramenta LSI levar em consideração no momento de calcular a similaridade entre os artefatos também ponderar as questões referentes ao estilo.

No exemplo seguinte o código do aluno foi transformado na sua respectiva representação no espaço vetorial através da aplicação das regras acima. É interessante observar que os caracteres de tabulação aparecem no vetor termo com seus respectivos números na tabela ASCII. O vetor termo na figura abaixo está na coluna representada pela palavra de origem inglesa “tokens”. A coluna chamada “18.irobo” representa o artefato e suas respectivas ocorrências de cada termo. A tabela abaixo não possui o fator de correção aplicado. Isto acontece porque o fator

somente pode ser computado quando toda a coleção de documentos já tiver sido processada. O exemplo abaixo (Figura 20) somente possui um único artefato e foi computado para fins de demonstração e legibilidade de apresentação do método.



The screenshot shows a window titled "Matrix Tokens x Código Fonte". It has three tabs: "Termos", "LSA", and "Print". The "Termos" tab is active, displaying a table with two columns: "Tokens" and "18.irobo". The table lists various tokens and their corresponding values. To the right, a code editor window titled "Código Fonte-18.irobo" displays a procedure definition in a programming language.

Tokens	18.irobo
POSIX [:space:] {10}	10.0
para	1.0
repetir	1.0
)	5.0
(5.0
POSIX [:space:] {9}	7.0
#	1.0
procedure	1.0
}	2.0
{	2.0
x	1.0
virarDireita	1.0
irAteX	2.0
o	1.0
POSIX [:space:] {32}	5.0
chegar	1.0
avancar	1.0
2	2.0
ate	1.0

```
procedure irAteX()
{
    repetir(2)
    {
        virarDireita()
        avançar(2)
    }
}
#para chegar ate o x
irAteX()
```

Figura 20 - Exemplo de decomposição do programa em sua forma algébrica

3.7 O FATOR DE AJUSTE

Os trabalhos que utilizam a técnica LSI geralmente preconizam a aplicação do fator de correção para melhorar o desempenho do método. Neste trabalho será utilizado o fator de correção por entropia que propõe a redução da importância dos termos mais freqüentes e o aumento dos menos freqüentes em todo o documento (KONTOSTATHIS, 2007):

$$g_i = 1 + \sum_j \frac{p_{ij} \log p_{ij}}{\log n}$$

$$a_{ij} = g_i \log(\text{tf}_{ij} + 1)$$

A literatura também registra que a aplicação do fator de correção causa grande impacto nos valores gerados pela técnica LSI. Algumas comparações utilizando a amostra recolhida são citadas abaixo:

- a. Utilizando o exercício 11 do aluno 12 as seguintes observações mostram que o fator de ajuste foi capaz de ressaltar semelhanças:

Aluno	Fator não Aplicado	Fator Entropia	Aluno	Fator não Aplicado	Fator Entropia
44	0,0538564491	0,8670410595	20	-0,1866327809	0,1861837652
30	0,1880050042	0,9294306123	36	-0,1446739258	0,1636862776
19	0,3148145795	0,8253536687	1	0,0260104932	0,2644201355
34	0,4502778474	0,9512214469	37	-0,0344232776	0,1847183394
48	-0,3675837811	0,1231039089	29	-0,0890663200	0,1038885680
3	-0,0458100414	0,3447654001	46	0,6882241711	0,8786103667
22	-0,0458100414	0,3447654001	32	0,3026213552	0,4904656492
40	-0,0458100414	0,3447654001	2	0,3164830176	0,4905760611

Figura 21 - Comparação dos trabalhos do aluno 12 no exercício 11 com os outros alunos usando dimensão k=10 e variando o fator de Entropia

F:\...ico\social network\data_experimento\exercicios\ex11\44. robo	F:\...ico\social network\data_experimento\exercicios\ex11\12. robo
<pre> repetir(2){ } virarDireita()) </pre>	<pre> repetir(10){ medirAgua(qtdAgua) se(qtdAgua<=40){ lembrar(reporqtdAgua,55-qtdAgu irrigar(reporqtdAgua)} dormir(6)} meiaVolta() procedure meiaVolta(){ virarDireita() } </pre>
<pre> repetir(6){ medirAgua(qtdAgua) se(qtdAgua < 40) { lembrar(reporqtdAgua,55 - qtdagu irrigar(reporqtdAgua)} dormir(10)) </pre>	

Figura 22 - Comparação visual entre os trabalhos dos alunos 12 e 44 no exercício 11: Usando o fator de entropia os documentos são apontados como semelhantes.

F:\...ico\social network\data_experimento\exercicios\ex11\30. robo	F:\...ico\social network\data_experimento\exercicios\ex11\12. robo
<pre> # Verificar e regar. </pre>	<pre> repetir(10){ medirAgua(qtdAgua) se(qtdAgua<=40){ lembrar(reporqtdAgua,55-qtdAgu irrigar(reporqtdAgua)} dormir(6)} </pre>
<pre> meiaVolta() repetir(10){ medirAgua(qtdAgua) se(qtdAgua < 40) { irrigar(55-qtdAgua)} dormir(6)) </pre>	<pre> meiaVolta() </pre>
<pre> procedure meiaVolta() { virarDireita() virarDireita() } </pre>	<pre> procedure meiaVolta() { virarDireita() virarDireita() } </pre>

Figura 23 - Comparação visual entre os trabalhos dos alunos 12 e 30 no exercício 11: Usando o fator de entropia os documentos são apontados como semelhantes.

3.8 ESCOLHA DA DIMENSÃO

A determinação da quantidade de valores singulares utilizados na redução da matriz é outro componente importante da técnica LSI. Em recuperação de informações é consenso entre os autores a manutenção de 200 até 500 valores singulares para a reconstrução da matriz. Isto acontece porque a quantidade de documentos e termos são superiores a milhões de unidades. Todavia nas aplicações da LSI relacionadas aos artefatos de construção do software é aconselhável segundo KUHN et. al. (2006) usar uma dimensão menor. Valores situados entre 10 até 25 dimensões estariam mais apropriados ao tamanho e a natureza do código fonte. No trabalho citado os autores sugeriram empiricamente que k poderia ser determinado por:

$$K = (m \times n)^{0,2}$$

Abaixo (Figura 24) os histogramas do exercício 1 para valores de k variando entre 2 até 25. Nestes histogramas é possível perceber que quando k atinge 10 dimensões existe uma clara distribuição das similaridades entre os artefatos produzidos pelos alunos em forma de curva normal. A partir de 10 dimensões a distribuição mostra a maioria dos trabalhos como similares e concentra a representação ao redor do valor 0.

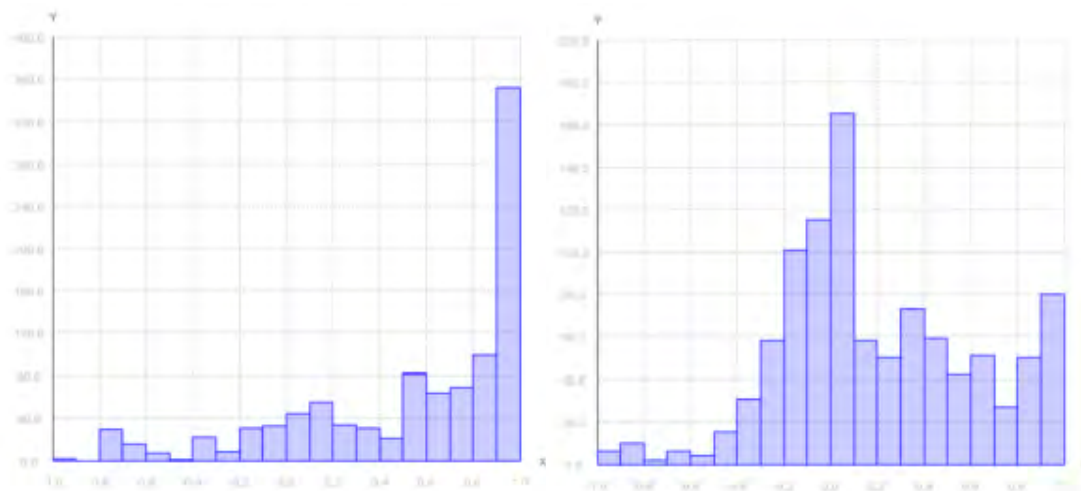


Figura 24 - À esquerda distribuição com $k = 2$ e a direita com $k = 5$

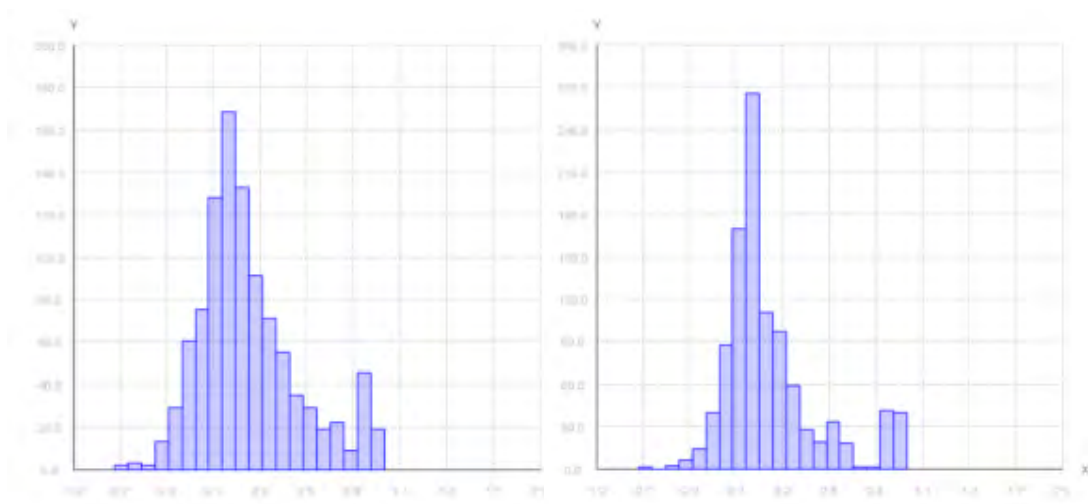


Figura 25 - À esquerda distribuição com $k = 10$ e a direita com $k = 15$

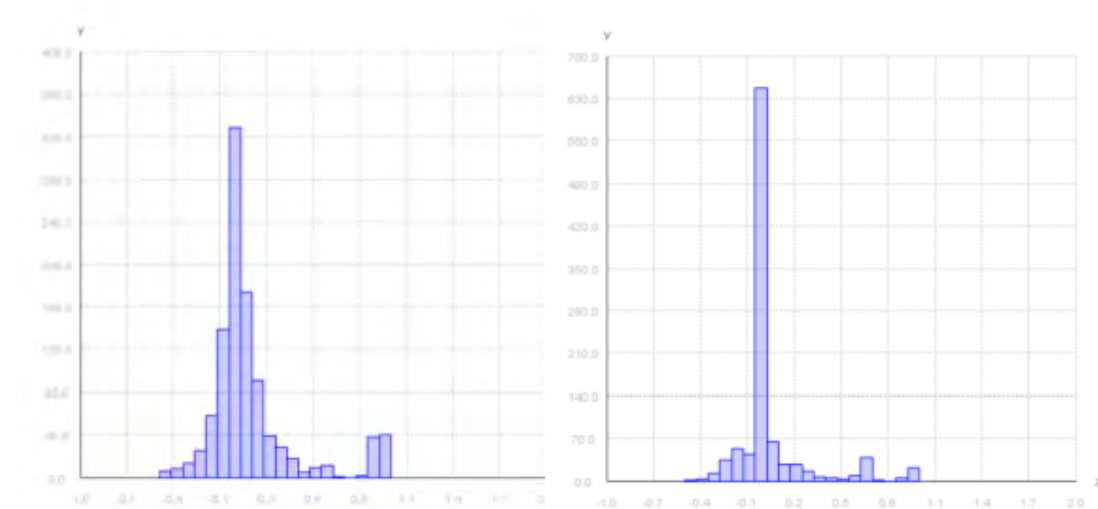


Figura 26 - À esquerda distribuição com $k = 20$ e a direita com $k = 25$

Uma avaliação empírica do grau de similaridade entre os artefatos mostrou que $k = 10$ é uma escolha adequada para representar o corpus coletado.

Todos os exercícios entregues foram submetidos ao processo de LSI e os dados gerados estão no anexo 5.

3.9 CÁLCULO DO QAP

Para todos os grupos de exercícios entregues foi aplicada a técnica LSI com $K=10$ e fator de ajuste pelo método da entropia. Esta técnica gerou 15 matrizes, uma para cada exercício. Em cada uma destas matrizes E o elemento e_{ij} representa o valor da similaridade entre o artefato do aluno i e do aluno j .

Este trabalho interpreta que as similaridades em cada matriz E não acontecem aleatoriamente, mas é uma possível expressão da relação entre o aluno i e o aluno j . Em decorrência desta interpretação todas as similaridades superiores a um valor de corte c , ou seja, e_{ij} maior que c , não ocorreriam isoladamente da rede social do aluno. Com o objetivo de verificar esta hipótese é necessário estabelecer a correlação entre os elementos e_{ij} com a rede social levantada através do questionário sociométrico. O questionário sociométrico representa a visão dos alunos sobre seus relacionamentos em sala de aula é uma medida direta da rede social. A representação do questionário sociométrico através da matriz S permite calcular a correlação entre s_{ij} e e_{ij} , para e_{ij} maior que c .

Nos casos de correlação de variáveis contínuas será utilizado o teste de Person cuja equação é dada por:

$$\rho = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^n (x_i - \bar{x})^2} \cdot \sqrt{\sum_{i=1}^n (y_i - \bar{y})^2}} = \frac{\text{cov}(X, Y)}{\sqrt{\text{var}(X) \cdot \text{var}(Y)}}$$

$$\bar{y} = \frac{1}{n} \cdot \sum_{i=1}^n y_i \quad \bar{x} = \frac{1}{n} \cdot \sum_{i=1}^n x_i$$

Contudo a literatura registra que os métodos de correlação não se aplicam diretamente as redes sociais devido a fato das $n(n-1)$ observações entre os n atores não serem consideradas independentes. Esta dependência implica mudanças significativas das medidas estatísticas do erro padrão e do valor-p (KRACKHARDT, 1988).

Uma solução para este problema que testa a hipótese nula para a similaridade entre duas redes formada pelos mesmos atores porem com relações diferentes é dada pela técnica QAP. A técnica QAP determina o erro padrão e o

valor-p através do cálculo da distribuição das correlações entre A e as possíveis permutações de B. A idéia básica deste procedimento é comparar a correlação entre os dados pesquisados contra uma distribuição aleatória das relações.

Para exemplificar os métodos as duas estruturas simples abaixo adaptadas do trabalho de KRACKHARDT (1987) serão utilizadas. Esta rede contendo somente 5 atores demonstra duas relações hipotéticas distintas: as de amizade que são representadas pela matriz A e de aconselhamento que são representadas pela matriz B (Figura 27). Neste exemplo deseja-se testar se a procura por conselhos acontece entre os atores que estão conectados através de relações de amizade. Em outras palavras procura-se determinar se as duas estruturas são similares.

$$A = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 \end{pmatrix} \quad B = \begin{pmatrix} 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 & 0 \end{pmatrix}$$

Figura 27 – Matrizes representando dois tipos de relacionamentos distintos

A aplicação da técnica QAP permite verificar que a distribuição das correlações utilizando o teste de Person entre a matriz A e todas as 120 possíveis permutações da matriz B são (Tabela 3):

Media de Similaridade	Número de Permutações	Percentual
0,8165_	4	3,30%
0,4082_	32	26,70%
0,0000_	48	40,00%
-0,4082_	32	26,70%
-0,8165_	4	3,30%
Total 120		100,00%

Tabela 3 – Distribuição de Similaridades

Dado que a correlação pelo teste de Person entre a matriz A e a matriz B originais é de $r=0,8165$, é possível perceber pela distribuição que a probabilidade desta relação acontecer ao acaso é de somente 3,3%. Com base nestas medidas é possível afirmar que a correlação de 0,8165, correlação forte pela escala de Person, é significativa a um nível de 0,033 dado a hipótese nula da distribuição aleatória.

Contudo em redes sociais que expressam problemas reais a quantidade de atores torna o método de permutar todas as combinações possíveis da matriz B intratável computacionalmente. Nestes casos é adotado o método analítico desenvolvido por Hubert (1976) que identificou propriedades específicas sob a premissa da distribuição normal que reduzem a necessidade de computação.

Antes do cálculo QAP o sociograma S sofreu três transformações para que a comparação fosse executada:

- a. A rede social foi reduzida a uma rede onde somente permaneceram os relacionamentos recíprocos, ou seja, ambos os alunos indicaram-se para fazer parte da mesma equipe;
- b. Para determinar a importância da relação entre i e j foi calculada para cada elemento de S a quantidade de nós que deveriam ser desconectados para que não existisse nenhum caminho entre o aluno i e j . Este processo permitiu quantificar a importância de cada relação e resultou na matriz S' ;
- c. A matriz S' foi então normalizada utilizando como referência o maior valor encontrado entre os elementos s'_{ij} .

Após este procedimento foi feita a aplicação do QAP entre todas as representações LSI dos trabalhos entregues pelos alunos e o sociograma S' utilizando-se como parâmetro do algoritmo o valor de 5000 permutações (tabela 4 e 5).

QAP Correlations

	SOCIOGRAMA-F	EX1	EX2	EX3	EX4	EX5	EX6	EX7	EX8
SOCIOGRAMA-FC-Normed	1.0000000000	0.0746962279	0.0675039366	0.0875991955	0.1299687624	0.0648619607	0.0462964959	0.0947453082	0.0934320539
EX1	0.0746962279	1.0000000000	0.4868603051	0.3417766392	0.2855650187	0.4012296498	0.1766486466	0.1118184254	0.1348517239
EX2	0.0675039366	0.4868603051	1.0000000000	0.3971283734	0.3716901243	0.4691205919	0.1718639135	0.1376439035	0.1641696990
EX3	0.0875991955	0.3417766392	0.3971283734	1.0000000000	0.4137783051	0.4629603624	0.1683373600	0.1357297301	0.1807169020
EX4	0.1299687624	0.2855650187	0.3716901243	0.4137783051	1.0000000000	0.5001285076	0.1692265868	0.1463529468	0.3181200925
EX5	0.0648619607	0.4012296498	0.4691205919	0.4629603624	0.5001285076	1.0000000000	0.2871410549	0.1106321961	0.1733718365
EX6	0.0462964959	0.1766486466	0.1718639135	0.1683373600	0.1692265868	0.2871410549	1.0000000000	0.1089261398	0.1416454017
EX7	0.0947453082	0.1118184254	0.1376439035	0.1357297301	0.1463529468	0.1106321961	0.1089261398	1.0000000000	0.1136865392
EX8	0.0934320539	0.1348517239	0.1641696990	0.1807169020	0.3181200925	0.1733718365	0.1416454017	0.1136865392	1.0000000000
EX9	0.1889877021	0.0952151641	0.1205873191	0.0856881067	0.1385311931	0.1737224609	0.1937903613	0.1509122849	0.2473823428
EX10	0.1087046862	0.2441095114	0.1893478632	0.2609660923	0.1747261137	0.1291793734	0.1158948764	0.0609875023	0.1336218417
EX11	0.2332492322	0.0670221522	0.0560388491	0.0291496813	0.0845471025	0.0288011786	0.0961702188	0.1631104648	0.0743950605
EX12	0.3108754256	0.1497882009	0.1315302253	0.1040261090	0.1386316717	0.0830315128	0.1179356277	0.1425853074	0.1536550075
EX13	0.3546144962	0.0679179172	0.1126198098	0.2771709263	0.3193312883	0.2102778405	0.1066614762	0.0843943134	0.2958500385
EX14	0.1569197923	0.0529954024	0.1061394834	0.0475367010	0.1561502814	0.1860428900	0.1508087963	0.1415951401	0.1653842181
EX15	0.3556511402	0.1496235132	0.1003756300	0.1621894538	0.2033682764	0.1932012290	0.1983109415	0.0920526087	0.1626980982
EX16	0.3066338301	0.1306025982	0.0811956897	0.1354894787	0.1075013876	0.1502771229	0.1127917320	0.1178016067	0.2110144943

QAP F-Values

	SOCIOGRAMA-F	EX1	EX2	EX3	EX4	EX5	EX6	EX7	EX8
SOCIOGRAMA-FC-Normed	0.0000000000	0.0809838027	0.1021795645	0.0525894836	0.0083983205	0.1079784036	0.1581683606	0.0269946009	0.0291941613
EX1	0.0809838027	0.0000000000	0.0001999600	0.0001999600	0.0011997600	0.0001999600	0.0045990800	0.0491901636	0.0335932821
EX2	0.1021795645	0.0001999600	0.0000000000	0.0001999600	0.0001999600	0.0001999600	0.0079984004	0.0277944412	0.0179964006
EX3	0.0525894836	0.0001999600	0.0001999600	0.0000000000	0.0001999600	0.0001999600	0.0061987601	0.0295940805	0.0099980002
EX4	0.0083983205	0.0011997600	0.0001999600	0.0001999600	0.0000000000	0.0001999600	0.0098980801	0.0169966016	0.0001999600
EX5	0.1079784036	0.0001999600	0.0001999600	0.0001999600	0.0001999600	0.0000000000	0.0003999200	0.0571885630	0.0145970806
EX6	0.1581683606	0.0045990800	0.0079984004	0.0061987601	0.0098980801	0.0003999200	0.0000000000	0.0305938814	0.0091981599
EX7	0.0269946009	0.0491901636	0.0277944412	0.0295940805	0.0169966016	0.0871885630	0.0305938814	0.0000000000	0.0379924029
EX8	0.0291941613	0.0335932821	0.0179964006	0.0099980002	0.0001999600	0.0145970806	0.0091981599	0.0379924029	0.0000000000
EX9	0.0007998400	0.0639872029	0.0339932032	0.0827834457	0.0185962804	0.0081983600	0.0003999200	0.0051989602	0.0001999600
EX10	0.0175964814	0.0019996001	0.0151969604	0.0027994402	0.0181963611	0.0601879619	0.0403919220	0.1629674137	0.0367926396
EX11	0.0001999600	0.1387722492	0.1831633747	0.3075384796	0.0835832804	0.3089382052	0.0305938814	0.0025984801	0.0939812064
EX12	0.0001999600	0.0073985201	0.0159968007	0.0431913622	0.0109978002	0.0761847648	0.0123975202	0.0047990400	0.0031993601
EX13	0.0001999600	0.1681663692	0.0661867633	0.0005998800	0.0001999600	0.0085982801	0.0375924818	0.0855828822	0.0003999200
EX14	0.0011997600	0.1723655313	0.0385922827	0.1973605305	0.0047990400	0.0023994801	0.0023995200	0.0043991203	0.0021995602
EX15	0.0001999600	0.0105978800	0.0603879206	0.0071985605	0.0009988000	0.0027994402	0.0003999200	0.0435912833	0.0029994000
EX16	0.0001999600	0.0145970806	0.0779844001	0.0145970806	0.0299940016	0.0099980002	0.0119976001	0.0125974808	0.0003999200

Tabela 4 – Resultado do QAP para todos os trabalhos

QAP Correlations

	EX9	EX10	EX11	EX12	EX13	EX14	EX15	EX16
SOCIOGRAMA-PC-Normed	0.1889877021	0.1087046862	0.2332492322	0.3105754256	0.3546144962	0.1569197923	0.3556511402	0.3066338301
EX1	0.0952151641	0.2441095114	0.0670221522	0.1497882009	0.0679173172	0.0529954024	0.1496235132	0.1306025982
EX2	0.1205873191	0.1893478632	0.0560388491	0.1315302253	0.1126198098	0.1061334834	0.1003756300	0.0811956897
EX3	0.0856881067	0.2609660923	0.0291496813	0.1040261090	0.2771709263	0.0475367010	0.1621894538	0.1354894787
EX4	0.1385311931	0.1747261137	0.0845471025	0.1386316717	0.3193312883	0.1561502814	0.2033682764	0.1075013876
EX5	0.1737224609	0.1291793734	0.0288011786	0.0830315128	0.2102778405	0.1860428900	0.1932012290	0.1502771229
EX6	0.1937903613	0.1158948764	0.0961702168	0.1179356277	0.1066614762	0.1508087963	0.1983109415	0.1127917320
EX7	0.1509122849	0.0609875023	0.1631104648	0.1425853074	0.0843943134	0.1415951401	0.0920526087	0.1178016067
EX8	0.2473823428	0.1336218417	0.0743950605	0.1536550075	0.2958500385	0.1653942181	0.1626930982	0.2110144943
EX9	1.0000000000	0.1649978608	0.0769397393	0.2634480000	0.1807922870	0.2356982082	0.2224063575	0.2513061464
EX10	0.1649978608	1.0000000000	0.2026936263	0.2610303164	0.2313825041	0.1820104420	0.3253050148	0.2215738446
EX11	0.0769397393	0.2026936263	1.0000000000	0.2749619186	0.1562006772	0.1421198398	0.2571669519	0.1829350889
EX12	0.2634480000	0.2610303164	0.2749619186	1.0000000000	0.1807910949	0.2336658090	0.3557051420	0.2208048552
EX13	0.1807922870	0.2313825041	0.1562006772	0.1807910949	1.0000000000	0.2421264946	0.3702283502	0.3064863384
EX14	0.2356982082	0.1820104420	0.1421198398	0.2336658090	0.2421264946	1.0000000000	0.3556363583	0.2605687976
EX15	0.2224063575	0.3253050148	0.2571669519	0.3557051420	0.3702283502	0.3556363583	1.0000000000	0.3559096158
EX16	0.2513061464	0.2215738446	0.1829350889	0.2208048552	0.3064863384	0.2605687976	0.3559096158	1.0000000000

QAP F-Values

	EX9	EX10	EX11	EX12	EX13	EX14	EX15	EX16
SOCIOGRAMA-PC-Normed	0.0007998400	0.0175964814	0.0001999600	0.0001999600	0.0001999600	0.0011997600	0.0001999600	0.0001999600
EX1	0.0639872029	0.0019996001	0.1387722492	0.0073985201	0.1681663692	0.1723655313	0.0105978800	0.0145970806
EX2	0.0339932032	0.0151969604	0.1831633747	0.0159968007	0.0661867633	0.0385922827	0.0603879206	0.0779844001
EX3	0.0827834457	0.0027994402	0.3075384796	0.0431913622	0.0005998800	0.1973605305	0.0071985605	0.0145970806
EX4	0.0185962804	0.0181963611	0.0835832804	0.0109978002	0.0001999600	0.0047990400	0.0009998000	0.0299940016
EX5	0.0081983600	0.0601879619	0.3089382052	0.0761847645	0.0085992801	0.0025994801	0.0027994402	0.0099980002
EX6	0.0003999200	0.0403919220	0.0305938814	0.0123975202	0.0375924818	0.0023995200	0.0003999200	0.0119976001
EX7	0.0051989602	0.1629674137	0.0025994801	0.0047990400	0.0855828822	0.0043991203	0.0435912833	0.0125974808
EX8	0.0001999600	0.0367926396	0.0939812064	0.0031993601	0.0003999200	0.0021995602	0.0029994000	0.0003999200
EX9	0.0000000000	0.0075984802	0.0625874847	0.0001999600	0.0021995602	0.0003999200	0.0001999600	0.0001999600
EX10	0.0075984802	0.0000000000	0.0007998400	0.0001999600	0.0017996401	0.0025994801	0.0001999600	0.0001999600
EX11	0.0625874847	0.0007998400	0.0000000000	0.0001999600	0.0051989602	0.0033993202	0.0001999600	0.0005998800
EX12	0.0001999600	0.0001999600	0.0001999600	0.0000000000	0.0019996001	0.0001999600	0.0001999600	0.0001999600
EX13	0.0021995602	0.0017996401	0.0051989602	0.0019996001	0.0000000000	0.0001999600	0.0001999600	0.0001999600
EX14	0.0003999200	0.0025994801	0.0033993202	0.0001999600	0.0001999600	0.0000000000	0.0001999600	0.0001999600
EX15	0.0001999600	0.0001999600	0.0001999600	0.0001999600	0.0001999600	0.0001999600	0.0000000000	0.0001999600
EX16	0.0001999600	0.0001999600	0.0005998800	0.0001999600	0.0001999600	0.0001999600	0.0001999600	0.0000000000

Tabela 5 – Resultado do QAP para todos os trabalhos

3.10 RESULTADOS

Uma análise inicial dos valores das tabelas 4 e 5 sugere uma correlação que sustenta a hipótese H1. Estes valores permitem sustentar a implicação entre o plágio P e as relações da rede social S.

A distribuição média das correlações apontadas pela técnica QAP estão no gráfico das figuras 28 e 29 abaixo e mostram uma correlação positiva de 0,16 em média.

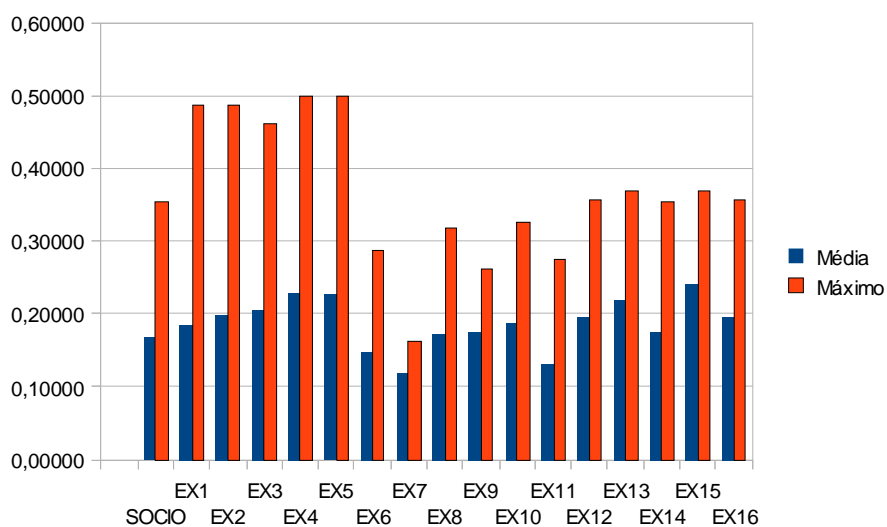


Figura 28 – Distribuição das Correlações

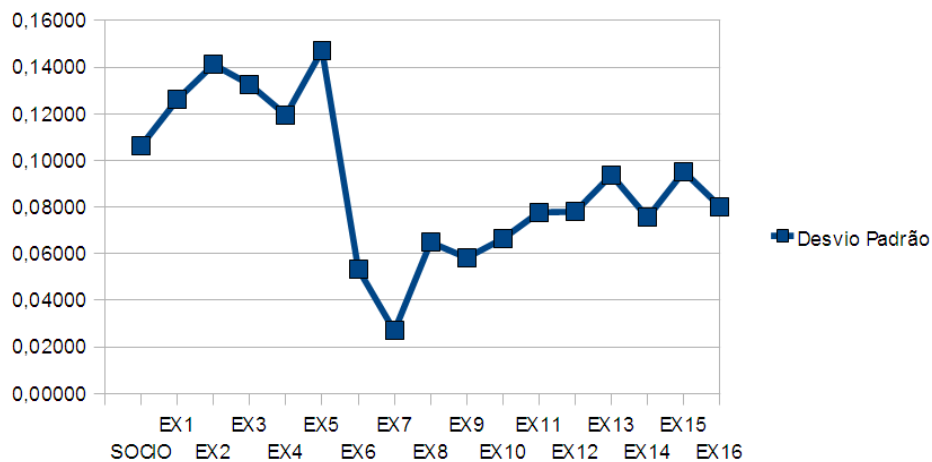


Figura 29 – Desvio Padrão das Correlações

Contudo devido às limitações de tempo não foi possível desenvolver novos experimentos para determinar se a relação de equivalência proposta na hipótese H2 é possível. A hipótese H2 sustenta que se existe plágio entre os artefatos dos alunos a e b , dado por $P(a,b)$, e existe um vínculo social entre os alunos a e b , dado por $S(a,b)$, então seria verdade que $P(a,b)$ equivale a $S(a,b)$. Um resultado de consequência prática e imediata desta propriedade é que o professor poderia ter conhecimento da rede social formada em sala de aula utilizando unicamente o código fonte entregue por seus alunos. Sendo o código fonte dos exercícios um artefato que já é produzido regularmente nas disciplinas de programação, o professor poderia modelar a rede social sem o ônus de aplicar, gerenciar e sintetizar os questionários sociométricos.

A mesma restrição de tempo também inviabilizou o teste da hipótese H3 onde está a oportunidade de aplicar o estudo. A hipótese H3 instrumentaliza o conhecimento e enfoca os professores e o processo de aprendizagem.

Nos trabalhos relacionados à hipótese H3 estão contidas as oportunidades para a criação de procedimentos concretos que permitam usar os dados da rede social para atuar positivamente sobre o resultado da aprendizagem dos alunos. Estas também abrem espaço para desenvolver técnicas que possibilitem medir e regular de forma concreta a colaboração entre os alunos através de artefatos que simbolizam a topologia da rede social.

4. CONCLUSÃO

Esta pesquisa apresentou uma abordagem inovadora para tratar a questão do plágio em exercícios de programação. A modelagem do plágio como evento social coloca uma nova perspectiva para lidar com este tema nos ambientes de ensino. A associação do plágio com a rede social permitiu também a aplicação de princípios de redes sociais e o desenvolvimento de uma nova utilização para as funções de plágio. Estes esforços foram consolidados em um experimento onde foi possível verificar a ocorrência da hipótese H1 nos dados empíricos extraídos de uma amostra de alunos. Dado as limitações de tempo não foi possível testar as hipóteses H2 e H3. A verificação destas hipóteses ficará a cargo de trabalhos futuros.

4.1 TRABALHOS FUTUROS

O trabalho de pesquisa proposto está situado em uma área multidisciplinar entre as ciências sociais e cognitivas e a disciplina de computação. Isto torna o processo metodológico difícil de ser equalizado levando em consideração as abordagens distintas entre as várias disciplinas.

É conhecido em ciências sociais que existem vários fatores humanos que podem afetar o comportamento dos indivíduos. Existem variáveis intervenientes nesta pesquisa que podem ter um controle mais apurado, entre elas pode-se citar: qual a influência da metodologia de apresentação da disciplina, o efeito da abordagem e desempenho didático do professor, a relevância da cultura acadêmica relativas à população de alunos e ao curso em questão que está originando os dados.

Neste sentido a abordagem estatística também pode ser aprimorada. A natureza exploratória deste trabalho justifica uma amostragem reduzida e com adequada relação de custo de obtenção. Contudo qualquer exploração futura dos tópicos deste trabalho deve criar condições de aumentar a amostragem e criar uma estrutura de grupos de controle.

Outra sugestão importante para os trabalhos futuros advém de uma análise do valor obtido para o nível de correlação frente à escala de Person. Na escala de correlação apontada pela técnica de Person é considerada uma correlação fraca os índices entre 0,0 e 0,5. A comparação do sociograma com todos os trabalhos mostra que a correlação média encontrada não ultrapassa 0,16. Essa informação permite extrapolar algumas considerações sobre a natureza metodologia do processo de correlação que podem se úteis em trabalhos futuros:

- a. A rede social dos parceiros de projeto não apresenta todas as relações sociais existentes em sala de aula. Outras relações sociais não capturadas pelo sociograma poderiam estar conduzindo a troca de informações. É possível perceber que quando os trabalhos são comparados entre si e não contra o sociograma os valores de correlação produzidos são significativamente maiores do que a média dos trabalhos comparados isoladamente com o sociograma (Tabela 6).

	SOCIO	EX1	EX2	EX3	EX4	EX5
SOCIO	1,00000	0,07470	0,06750	0,08760	0,12997	0,06486
EX1	0,07470	1,00000	0,48686	0,34178	0,28557	0,40123
EX2	0,06750	0,48686	1,00000	0,39713	0,37169	0,46912
EX3	0,08760	0,34178	0,39713	1,00000	0,41378	0,46296
EX4	0,12997	0,28557	0,37169	0,41378	1,00000	0,50013
EX5	0,06486	0,40123	0,46912	0,46296	0,50013	1,00000

Tabela 6 – Comparação das relações entre os trabalhos isoladamente

- b. A rede social é dinâmica, ou seja, a quantidade e os alunos que oferecem seus trabalhos ou parte das soluções variam ao longo dos exercícios. Desta forma a rede teria configurações diferentes ao longo da entrega dos trabalhos o que justificaria uma correlação fraca.

Estas observações sugerem que talvez uma medida mais adequada de correlação seja utilizar a rede do ego de cada aluno e não a rede inteira da sala. (rede do ego são todos os nós conectados a uma distância específica de um nó de interesse. Neste caso a uma distância igual a 1, conforme figura 30).

A utilização da correlação com base na rede do ego do aluno pode originar mais informações relevantes para a pesquisa como um todo e podem permitir encontrar valores mais importantes na escala de Person.

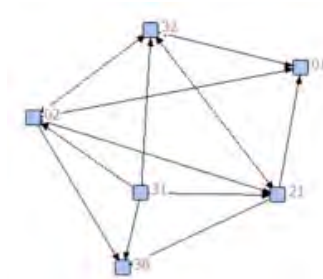


Figura 30 – Rede do Ego do Aluno 2

Um exemplo desta suspeita é dado a seguir para o aluno 2. O aluno 2 tem sua rede do ego dado na figura 30 onde mostra o relacionamento com os alunos: 32, 01, 31, 21 e 38.

No exercício número 4 é possível perceber que quando a similaridade é maior que 0,85 o trabalho do aluno 2 não está associado ao aluno 31. Mas o aluno 21 está presente (figura 31).

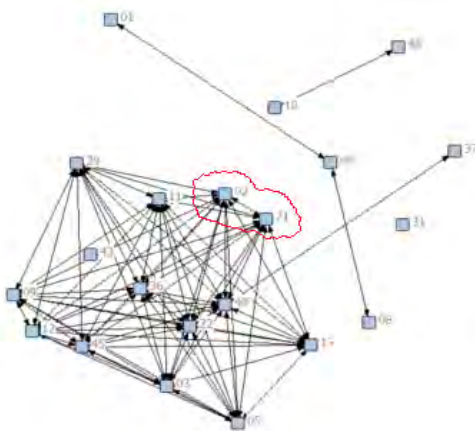


Figura 31 – Associação entre os alunos 2 e 21

No exercício número 7 é possível perceber que quando a similaridade é maior que 0,85 o trabalho do aluno 2 esta associado ao aluno 31 e ao aluno 21.(Figura 32)



Figura 32 – Associação entre os alunos 2, 31 e 21

No exercício número 8 é possível perceber que quando a similaridade é maior que 0,85 o trabalho do aluno 2 esta associado ao aluno 21 e ao aluno 1. Mas não está associado ao aluno 31 (Figura 33).



Figura 33 – Associação entre os alunos 2, 21 e 1

Os exemplos listados acima foram destacados somente para pontuar que um estudo futuro pode aprimorar o método através de um novo processo de correlação que permita levar em consideração esta flutuação que pode ter origem na disponibilidade dos colegas fornecedores de plágio em um determinado momento.

REFERÊNCIAS BIBLIOGRÁFICAS

ARULSELVAN, A.; COMMANDER, C.W.; ELEFTERIADOU, L.; PARDALOS, P.M. Detecting critical nodes in sparse graphs, *Computers and Operations Research*, 2008.

BARROS, J. P.; ESTEVENS, L.; DIAS, R.; PAIS, R.; SOEIRO, E. Using lab exams to ensure programming practice in an introductory programming course, In *Proceedings of the 8th Annual Conference on innovation and Technology in Computer Science Education*, ACM, 2003, p. 16-20.

BENNEDSEN, J.; CASPERSEN, M.E. Assessing Process and Product – A Practical Lab Exam for an Introductory Programming Course, Submitted for 36th Annual *Frontiers in Education Conference*, San Diego, USA, 2006.

BENNEDSEN, J.; CASPERSEN, M.E. Failure rates in introductory programming, *SIGCSE Bull*, ACM, 2007, v. 39, n. 2, p. 32-36.

BERZAL, F.; CUBERO, J.C.; JIMENEZ, A. Hierarchical program representation for program element matching, *LNCS*, Springer Berlin / Heidelberg , 2007, v. 4881, p. 467–476.

BILLS, D.; CANOSA, R. Sharing introductory programming curriculum across disciplines, *Proceedings of the SIGITE Conference*, 2007, p. 18-20.

BORGATTI, S. P. Identifying sets of key players in a network, *Computational and Mathematical Organization Theory*, 2006, v.12, p.21–34.

BOWYER, K.W.; HALL, L.O. Experience using “MOSS” to detect cheating on programming assignments, *Frontiers in Education Conference*, 1999, v.3, p. 13B3/18-13B3/22.

BURD, E.; BAILEY, J. Evaluating clone detection tools for use during preventative maintenance, In *Proc. of the Int'l Workshop on Source Code Analysis and Manipulation*, 2002, p.36-43.

CARROLL J.; APPLETON J. Plagiarism A Good Practice Guide, JISC report, 2001.

CASPERSEN, M.; KOLLING, M. STREAM: A First Programming Process, ACM Transactions on Computing Education, 2009, v.9, n. 1, p. 1-29.

CASSEL, L.; MCGETTRICK, A.; GUZDIAL, M.; ROBERTS, E. The current crisis in computing: what are the real issues?, In *Proceedings of the 38th SIGCSE Technical Symposium on Computer Science Education*, SIGCSE '07, ACM, 2007, p. 329-330.

CHEANG, B., KURNIA, A., LIM, A., OON, W.-C. On automated grading of programming assignments in an academic institution, *Computers and Education*, 2003, v. 41, p. 121–131.

CHURCHILL, E.F.; HALVERSON, C. A. Social Networks and Social Networking, *IEEE Internet Computing*, 2005, v. 9, n. 5, p. 14–19.

COSMA, G.; JOY, M. Towards a definition of source-code plagiarism. *Education*, *IEEE Transactions on*, 2008, v.51, n.2, p. 195–200.

CULWIN, F.; MACLEOD, A. LANCASTER, T. Source Code Plagiarism in UK HE Schools, Technical Report SBU-CISM-01, South Bank University, 2001.

DALY, C.; WALDRON, J. Assessing the assessment of programming ability, *Proceedings of the 35th SIGCSE technical symposium on Computer science education*, ACM Press, 2004, p. 210 – 213.

DALY, C., HORGAN, J. Patterns of plagiarism, *Proceedings of the 36th SIGCSE technical symposium on Computer science education*, 2005, p. 383 – 387.

DAWSON, S. A study of the relationship between student social networks and sense of community. *Educational Technology & Society*, 2008, v.11, n.3, p.224–238.

DAVEY, N.; BARSON, P.; FIELD, S.; FRANK, R.; TANSLEY, D. The development of a software clone detector. *International Journal of Applied Software Technology*, 1995, v.1, n.(3-4), p.219-236.

DEERWESTER, S.; DUMAIS, S. T.; FURNAS, G. W.; LANDAUER, T. K.; HARSHMAN, R. Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, 1990.

DEIBEL, K. Team Formation Methods for Increasing Interaction during In-Class Group Work, 10th annual SIGCSE Conference on Innovation and Technology in Computer Science Education, Portugal, 2005, p. 291-295.

DENNING, P. J.; MCGETTRICK, A. Recentering computer science. *ACM*, 2005, v. 48, n.11, p.15-19.

DENNING, P. J. The profession of IT Voices of computing, *ACM*, 2008, v. 51, n.8, p. 19-21.

DEY, S.; SOBHAN, A. Impact of unethical practices of plagiarism on learning, teaching and research in higher education: Some combating strategies, In 7th International Conference on Information Technology Based Higher Education and Training, 2006, p.388–393.

DEVLIN, M.; GRAY, K. In their own words: a qualitative study of the reasons Australian university students plagiarize, *Higher Education Research & Development*, 2007, v.26, n. 2, p. 181–198.

DICK, M.; SHEARD, J.; BAREISS, C.; CARTER, J.; JOYCE, D.; HARDING, T.; LAXER, C. Addressing student cheating: definitions and solutions, *SIGCSE Bull*, *ACM*, 2003, v.35, n.2, p. 172-184.

ENGELS, S.; LAKSHMANAN, V.; CRAIG, M. Plagiarism detection using feature-based neural networks, *SIGCSE Bull*, 2007, v.39, n.1, p. 34-38.

FRANTZESKOU, G.; GRITZALIS, S.; MACDONELL, S. Source code authorship analysis for supporting the cybercrime investigation process, *Proceedings of the First International Conference on E-business and Telecommunication Networks*, Kluwer Academic Publishers, 2004, p. 85–92.

FRANTZESKOU, G.; STAMATATOS, E.; GRITZALIS, S.; KATSIKAS, S. Effective identification of source code authors using byte-level information. In Proceedings of the 28th international Conference on Software Engineering ICSE '06, ACM,2006, p. 893-896.

FREIRE, M. Visualizing program similarity in the Ac plagiarism detection system, In Proceedings of the Working Conference on Advanced Visual interfaces, ACM, 2008, p.404-407.

GARCIA-MATEOS, G.; FERNANDEZ-ALÉMAN, J. L. Make Learning Fun with Programming Contests. In *Transactions on Edutainment II*, Lecture Notes In Computer Science, Springer-Verlag, 2009, v. 5660, p.246-257.

GABEL, M.; JIANG, L.; SU, Z. Scalable detection of semantic clones, In ICSE '08: Proceedings of the 30th international conference on Software engineering, ACM, 2008, p.321–330.

GITCHELL, D.; TRAN, N. A utility for detecting similarity in computer programs, Proceedings of 30th SCGCSE Technical Symposium,1998, p. 266-270.

GRANT, S.; CORDY,J.R. Vector space analysis of software clones. In Proc. ICPC 2009, 17th IEEE International Conference on Program Comprehension, 2009, p. 233-237.

GROSSMAN,D. A.; FRIEDER, O. Information Retrieval: Algorithms and Heuristics, 2nd edition. Springer,2004

HARRIS, J. K. Plagiarism in computer science courses, In *Proceedings of the Conference on Ethics in the Computer Age*, ACM, 1994, p. 133-135.

HEMMENDINGER, DAVID. The ACM and IEEE-CS Guidelines for Undergraduate CS Education. Communications of THE ACM, 2007, v. 50, n. 5.

HIGGINS, C. Automated assessment and experiences of teaching programming. ACM Journal on Educational Resources in Computing, 2005, v. 5, n. 3.

HWANG, C. J.; GIBSON, E. D. Using an Effective Grading Method for Preventing Plagiarism of Programming Assignments, ACM SIGCSE Bulletin, 1982 , n.14

HORWITZ, S.; REPS, T.; BINKLEY, D. Interprocedural slicing using dependence graphs. ACM, 1990, v.12, n.1, p.26-60.

HUISMAN, M.; VAN DUIJN, M.A.J. Software for statistical analysis of social networks, Models and methods in social network analysis, Cambridge University Press, 2005, p. 270–316.

JACOBSON, N. Using on-computer exams to ensure beginning students' programming competency, SIGCSE Bull, ACM, 2000, v.32, n.4, p. 53-56.

JAMALI, M.; ABOLHASSANI, H. Different Aspects of Social Network Analysis, In Proceedings of the 2006 IEEE/WIC/ACM international Conference on Web intelligence, IEEE Computer Society, 2006, p. 66-72.

JOHNSON, D. W., JOHNSON, R. T. Social interdependence theory and university instruction – Theory into practice. Swiss Journal of Psychology, 2002, v.61, p. 119–129.

JOY, M.; LUCK, M. Plagiarism in programming assignments. IEEE Transactions of Education, 1999, v. 42, n. 2, p. 129–133.

KANDA, T.; SATO, R.; SAIWAKI, N.; ISHIGURO, H. A Two-month Field Trial in an Elementary School for Long-term Human-robot Interaction, IEEE Transactions on Robotics (Special Issue on Human-Robot Interaction), 2007, v.23,n.5, p.962-971.

KÁTAIA, Z.; JUHÁSZB, K.; ADORJÁNIA, A. K. On the role of senses in education, Computers & Education, 2008, v. 51, n. 4, p. 1707-1717

KIMURA, M.; SAITO, K.; MOTODA, H. 2009. Minimizing the Spread of Contamination by Blocking Links in a Network, Association for the Advancement of Artificial Intelligence, 2008.

KLEINBERG, J. The convergence of social and technological networks. *Commun, ACM*, 2008, v. 51, n.11, p.66-72.

KOSCHKE, R. Identifying and Removing Software Clones, *Software Evolution*, Springer Verlag, 2008,p.15–39.

KONTOGIANNIS, K. Evaluation Experiments on the Detection of Programming Patterns Using Software Metrics, In *Proceedings Fourth Working Conference on Reverse Engineering*, 1997.

KONTOSTATHIS, A.; POTTENGER, W. M. Essential Dimensions of Latent Semantic Indexing (EDLSI). In *Proc. 40th Annual Hawaii Intl. Conf. on System Sciences (CD-ROM)*, 2007.

KONTOSTATHIS, A.; POTTENGER, W. M. A framework for understanding LSI performance. *Information Processing & Management*, 2006, 42(1), p. 56-73.

KRACKHARDT, D. QAP Partialling as a Test of Spuriousness. *Social Networks*, 1987, 9, 171-186.

KRACKHARDT, D. Predicting with networks: Nonparametric multiple regression analysis of dyadic data. *Soc. Networks*, 1988, 10, 359-381.

KUHN, A.; DUCASSE, S.; GIRBA, T. Semantic clustering: Identifying topics in source code. *Information and Software Technology*, 2006.

LISTER, R.; LEANEY, J. Introductory programming, criterion-referencing, and bloom, In *Proceedings of the 34th SIGCSE Technical Symposium on Computer Science Education*, ACM, 2003, p. 143-147.

LISTER, R. Computer Science Teachers as Amateurs, Students and Researchers. In *Proceedings of the 5th Baltic Sea Conference on Computing Education Research*, Finland, 2005, p. 3-12.

LIU, C.; CHEN, C.; HAN, J.; YU, P. S. GPLAG: detection of software plagiarism by program dependence graph analysis, In *Proceedings of the 12th ACM SIGKDD*

international Conference on Knowledge Discovery and Data Mining, ACM, 2006, p.872-881.

LU, J. J.; FLETCHER, G. H. Thinking about computational thinking, In Proceedings of the 40th ACM Technical Symposium on Computer Science Education, ACM, New York, NY, p. 260-264

LUJAN, H. L.; DICARLO, S. E. Too much teaching, not enough learning: what is the solution? *Adv Physiol Educ*, 2006; v.30, p.17–22.

LUQUINI, E. ; OMAR, N. From Source Code Plagiarism to Social Network. In: SIGCSE 2010 The 41st ACM Technical Symposium on Computer Science Education, 2010.

LUQUINI, E. ; OMAR, N. Repensando o Plágio no Ensino de Programação através de Redes Sociais. In: CIESC'2009 - 17th Iberian-American Conference on High Education in Computer Science, 2009, Pelotas. Proceedings of the CIESC'2009 - 17th Iberian-American Conference on High Education in Computer Science. Pelotas : SBC- CLEI - UFPL, 2009. v. 1. p. 1-2.

MANN, S.; FREW, Z. Similarity and originality in code: plagiarism and normal variation in student assignments, In Proceedings of the 8th Australian Conference on Computing Education, ACM International Conference Proceeding Series, Australian Computer Society, 2006, p. 143-150.

MARTÍNEZ-MONÉS, A.; DIMITRIADIS, Y.; RUBIA-AVI, B.; GÓMEZ-SÁNCHEZ, E.; FUENTE-REDONDO, P. Combining qualitative evaluation and social network analysis for the study of classroom social interactions, *Computers and Education*, 2003, v.41, n.4, p. 353-368.

MAURER, H.; KAPPE, F. ZAKA, B. Plagiarism - A Survey, *Journal of Universal Computer Science*, 2006, vol. 12, n. 8, p. 1050-1084.

MCCRACKEN, M.; ALMSTRUM, V.; DIAZ, D.; GUZDIAL, M.; HAGAN, D.; KOLIKANT, Y. B.; LAXER, C.; THOMAS, L.; UTTING, I.; WILUSZ, T. A multi-

national, multi-institutional study of assessment of programming skills of first-year CS students, SIGCSE Bull, 2001, v 33, n.4, p. 125-180.

MCGETTRICK, A.; BOYLE, R.; IBBETT, R.; LLOYD, J.; LOVEGROVE, G.; MANDER, K. Grand Challenges in Computing: Education--A Summary, The Computer Journal, 2005, v. 48, p. 42-48.

MILKA, P. Social Networks and The Semantic Web. New York - USA: Springer Science - Business Media, 2007.

MOZGOVOY, M. Desktop tools for offline plagiarism detection in computer programs, Informatics in education, 2006, v.5, n.1, p.97-112

NAKOV, P. Latent semantic analysis of textual data, In Proceedings of the Conference on Computer Systems and Technologies, Eds. CompSysTech '00, ACM, 2000, p. 5031-5035.

NAUDE, E.; HÖRNE, T. Cheating or collaborative work: Does it pay? Issues in Informing Science and Information Technology, 2006, n.3, p. 459-466.

NIEZGODA, S.; WAY, T. P. SNITCH: a software tool for detecting cut and paste plagiarism, In Proceedings of the 37th SIGCSE Technical Symposium on Computer Science Education - SIGCSE '06, ACM, 2006, p.51-55

OHNO, A.; MURAO, H. A Quantification of Students Coding Style Utilizing HMMBased Coding Models for In-Class Source Code Plagiarism Detection, In Proceedings of the 2008 3rd international Conference on innovative Computing information and Control, ICICIC. IEEE Computer Society, 2008

OPHIR, B. M. W.; BROWNE, M., Understanding Search Engines: Mathematical Modeling and Text Retrieval, Society for Industrial and Applied Mathematics, Philadelphia, 2005.

PATENAUDE, J.-F.; MERLO, E.; DAGENNAIS, M.; LAGUE, B. Extending software quality assessment techniques to java systems, In Proceedings of the 7th.

International Workshop on Program Comprehension. IWPC'99. IEEE Computer Society Press, 1999.

PEARS, A.; SEIDMAN, S.; MALMI, L.; MANNILA, L.; ADAMS, E.; BENNEDSEN, J.; DEVLIN, M.; PATERSON, J. A survey of literature on the teaching of introductory programming. SIGCSE Bulletin, ACM, New York – USA, 2007, v. 13, n. 2

PRECHELT, L.; MALPOHL, G.; PHILIPPSEN, M. Finding Plagiarisms among a Set of Programs with JPlag Java, Journal of Universal Computer Science, 2002, v. 8, n. 11, p. 1016-1038.

WAGNER, N. Plagiarism by Student Programmers.

<http://www.cs.utsa.edu/~wagner/pubs/plagiarism0.html>, acessado em 10 abril de 2009.

WINSLOW, L. E. Programming pedagogy – a psychological overview. SIGCSE Bulletin, 1996, v. 28, n. 3.

WISE, M. J. YAP3: Improved Detection of Similarities in Computer Program and Other Texts. ACM SIGCSE Bulletin, 1996, vol. 28, n. 1, p. 130-134.

ROSALES, F.; GARCIA, A.; RODRIGUEZ, S.; PEDRAZA, J.L.; MENDEZ, R.; NIETO, M.M. Detection of Plagiarism in Programming Assignments, IEEE Transactions on Education, 2008, v. 51, n. 2, p. 174-183

ROSETH, C.; JOHNSON, D.; JOHNSON, R. Promoting early adolescents' achievement and peer relationships: the effects of cooperative competitive and individualistic goal structures, Psychol Bull, 2008, v.1342, p.223–46.

ROY; C. K.; CORDY, J. R.; KOSCHKE, R. Comparison and Evaluation of Code Clone Detection Techniques and Tools: A Qualitative Approach. Science of Computer Programming, 2009, v.74, n.7.

ROY, C.K.; CORDY, J.R. NICAD: Accurate Detection of Near-Miss Intentional Clones Using Flexible Pretty-Printing and Code Normalization, Program Comprehension, 2008. ICPC 2008. The 16th IEEE International Conference on ,p.172-181.

RUDOLPH, H.; ;RADCLIFF, P.J. Anti-Plagiarism: a Weapon of Mass Self Delusion?, Proceedings of the 2007 AAEE Conference, 2007.

SARMA, A.; MACCHERONE, L.; WAGSTROM, P. ; HERBSLEB, J. Tesseract: interactive visual exploration of socio-technical relationships in software development, International Conference on Software Engineering, 2009, p. 23-33.

SCHLEIMER, S.; WILKERSON, D. S.; AIKEN, A. Winnowing: local algorithms for document fingerprinting, Proceedings of the 2003 ACM SIGMOD International Conference on Management of Data 2003, 2003, p. 76–85.

SPINELLIS,D.; ZAHARIAS, P.; VRECHOPOULOS, A. Coping with plagiarism and grading load: Randomized programming assignments and reflective grading, Computer Applications in Engineering Education, 2007.

SRAKA, D.; KAUCIC, B. Source code plagiarism, Information Technology Interfaces, 2009. ITI '09. Proceedings of the ITI 2009 31st International Conference on , 2009, p.461-466.

SHACKELFORD, R.; CASSEL, L.; CROSS, J.; DAVIES, G.; IMPAGLIAZZO, J.; KAMALI, R.; LAWSON, E.; LEBLANC, R.; MCGETTRICK, A.; SLONA, R.; TOPI, H.; VANVEEN, M. Computing Curricula 2005 - The Overview Report, Joint Task Force ACM, AIS, IEEE-CS, 2005.

SHEARD, J.; CARBONE, A; DICK, M. Determination of factors which impact on IT students' propensity to cheat. In Proceedings of the Fifth Australasian Conference on Computing Education, Australina Computer Society, 2003

SHEARD, J.; DICK, M.; MARKHAM, S.; MACDONALD, I.; WALSH, M. Cheating and plagiarism: perceptions and practices of first year IT students. Proc. 7th Annual Joint Conference on Innovation and Technology in Computer Science Education, 2002, p. 183-187.

SHEARD, J.; HAMILTON, M. S; L'ONNBERG, J. Analysis of research into the teaching and learning of programming. International Computing Education Research Workshop. ACM, 2009.

STAMATATOS, E. A survey of modern authorship attribution methods. *Journal of the American Society for Information Science and Technology*, 2006, v.60, n.3, p.538–556.

SIH, A.; HANSER, S. F.; MCHUGH, K. A. Social network theory: new insights and issues for behavioral ecologists. *Behavioral Ecology and Sociobiology*, 2009, v.63, p.975-988.

STEIN,B.; KOPPEL,M.; STAMATATOS,E. Plagiarism analysis, authorship identification,and near-duplicate detection pan'07. *SIGIR Forum*,2007, v.41, n.2, p. 68-71.

YADIN, A.; OR-BACH, R. Fostering individual learning: when and how, *SIGCSE Bull, ACM*, 2008, v.40, n.4, p. 83-86.

YARDI, S.; BRUCKMAN, A. What is computing?: bridging the gap between teenagers' perceptions and graduate students' experiences. In *Proceedings of the Third international Workshop on Computing Education Research ICER '07*, ACM, 2007, p. 39-50.

ZOBEL, J.; HAMILTON, M. Managing Student Plagiarism in Large Academic Departments. *Australian Universities Review*, 2002, p.119-126.

TAN, T. S.; CHEUNG, W. S. Effects of computer collaborative group work on peer acceptance of a junior pupil with attention deficit hyperactivity disorder (ADHD). *Computers & Education*, 2008, v.50, n.3, p.725-741.

VALENTE, T.W.; GALLAHER, P.; MOUTTAPA, M. Using social networks to understand and prevent substance use: a transdisciplinary perspective, *Substance Use & Misuse*, 2004, v. 39, p. 1685-1712.

VAMPLEW, P.; DERMOUDY, J. An anti-plagiarism editor for software development courses. In *Proceedings of the 7th Australasian Conference on Computing Education*, Australian Computer Society 2005, v.42

VYGOTSKY, L. S. Mind in society: The development of higher psychological processes , Cambridge Mass: Harvard University Press, 1978

ZHANG, C.; XU, Y. C.; ZHANG, C. How Do People Seek Knowledge in Information System Projects: A New Perspective from Social Network Theory, The Third International Multi-Conference on Computing in the Global Information Technology, 2008

ANEXOS

Anexo 1 – Ambiente Robowillie

Dentro do objetivo de produzir código fonte durante uma disciplina introdutória foi adotado o ambiente de iniciação à programação envolvendo a temática dos robôs. Esta abordagem já foi apresentada em vários trabalhos onde a hipótese principal é que a robótica seria um veículo para atingir efetivamente o público iniciante em programação. Contudo a implementação de kits de robótica não são sempre possíveis conforme apontado pelo autor Halma (2009). Este autor então propôs um ambiente de programação envolvendo a temática dos robôs com uma solução somente em software. Este ambiente de programação foi utilizado nos experimentos em classes de graduação em ciências da computação e no ensino médio feito pelos autores Vahldick e Benitti (2009) com resultados positivos.

Apesar dos autores Vahldick e Benitti (2009) terem criado uma variação do RoboMind para torná-lo apto a usar o conceito de memória e alguns operadores de decisão (RoboFurb,2009), este projeto adotou uma customização própria e mais abrangente. Entre as mudanças implementadas pode-se descrever:

- 1) O conceito de memória também foi implementado e está associado diretamente com um avaliador completo para expressões lógicas e relacionais integrado ao sistema. A adoção foi feita para tornar o processo de decisão compatível com um conjunto completo de operadores relacionais e lógicos com as mesmas capacidades oferecidas, mas sintaxes de linguagens como Java, C# e C++.
(FIGURA 1);

- 2) Foi adotado um visualizador para acompanhar as mudanças que acontecem na memória do robô, introduzindo o conceito de depuração de forma mais significativa. (FIGURA 2);

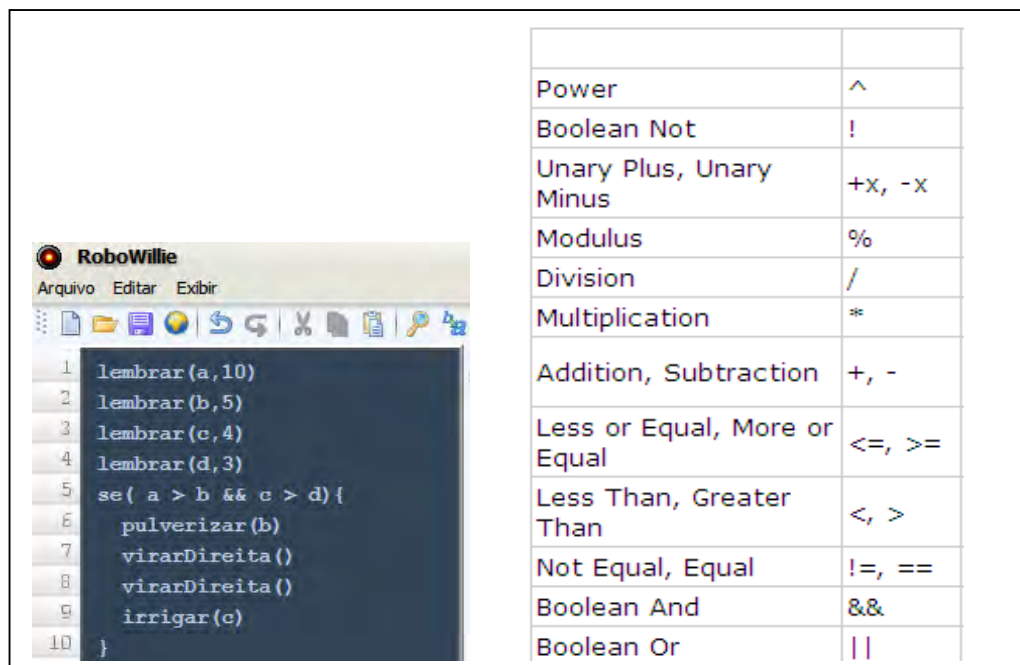


FIGURA 1 – Editor do ambiente e lista de operadores

The image shows the 'Memória do Robo' (Robot Memory) viewer. It displays a table with two columns: 'Variável' (Variable) and 'Valor' (Value). The table contains the following data:

Variável	Valor
b	5.0
a	10.0
d	3.0
c	4.0

FIGURA 2 – Visualizador da memória do Robô

3) Foi implementado no sistema a capacidade do robô trocar informações com os objetos do universo, proporcionando a criação de atividades mais complexas do que as existentes no RoboMind que são baseadas exclusivamente em desviar de objetos e mover obstáculos. Por exemplo, na figura abaixo o robô é capaz de medir as grandezas como água, fertilizante e pragas que acometem a planta a cada momento além de irrigá-las, fertilizá-las e pulverizá-las. A planta por sua vez executa uma simulação de seu ciclo de vida que é independente do robô. Isto permite que a

planta apareça em várias configurações ao longo do tempo em acordo com seu estado que é calculado dado os recursos que ela dispõe e taxas de consumo. (FIGURA 3);



FIGURA 3 – RoboMind expandido para incluir elementos de interação

4) O ambiente também ganhou um controle para reiniciar as atividades, na figura abaixo em vermelho. Esta função foi adotada para que o robô também se possuísse um ciclo de vida determinado pelo seu nível de bateria. Esta função foi implementada de forma que indiretamente as atividades a serem executadas possam ganhar a idéia de tempo e eficiência e novamente abrindo possibilidade de atividades mais ricas e interativas. (FIGURA 4)



FIGURA 4 – Inclusão de elementos de tempo e eficiência

Anexo 2 – Exercícios de Programação

Os seguintes exercícios foram solicitados ao longo da disciplina como atividades somativas e complementares a disciplina tradicional de introdução a programação e compreenderam 40% da nota final do aluno:

- 1) Movimentar o robô da posição inicial para a posição final; (FIGURA 1)

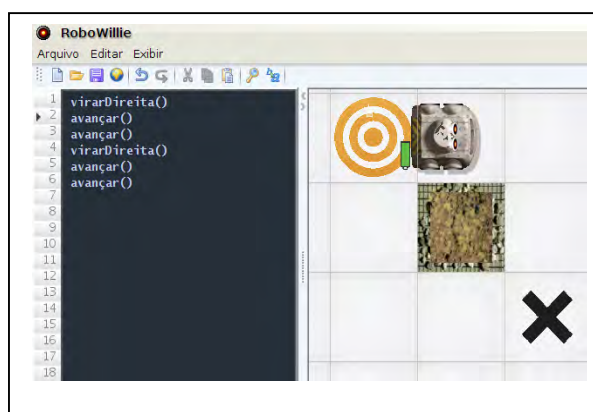


FIGURA 1 – Exemplo visual do exercício 1

- 2) Movimentar a caixa da sua posição inicial para posição final; (FIGURA 2)



FIGURA 2 - Exemplo visual do exercício 2

3) Movimentar as caixas da sua posição inicial para posição final; (FIGURA

3)

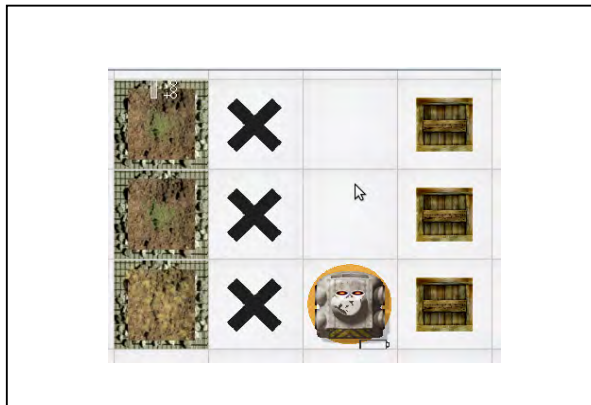


FIGURA 3 Exemplo visual do exercício 3

4) Faça o robô andar em círculos; (FIGURA 4)

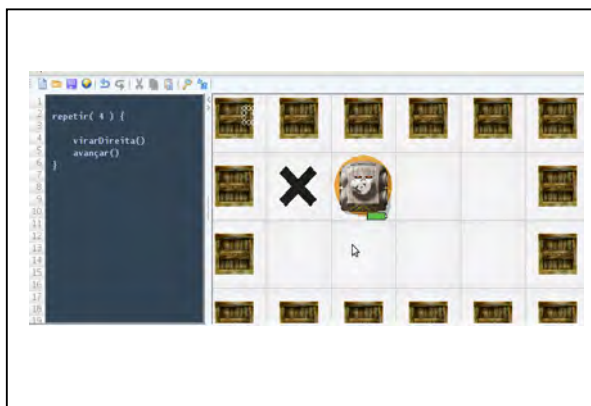


FIGURA 4 - Exemplo visual do exercício 4

5) Faça o robô sair do labirinto (FIGURA 5)



FIGURA 5 - Exemplo visual do exercício 5

6) Faça o robô mover as baterias para a posição indicada (FIGURA 6)

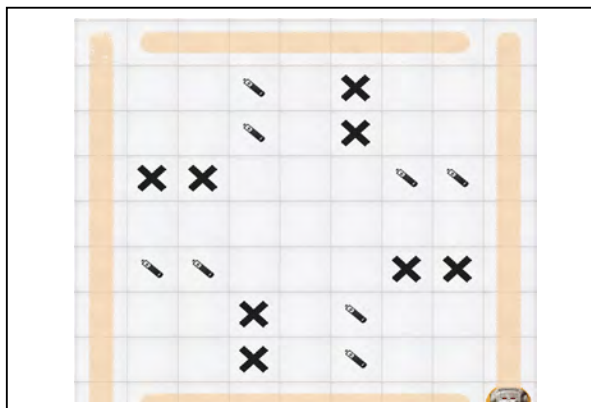


FIGURA 6 - Exemplo visual do exercício 6

7) Faça o robô verificar a quantidade de água de cada planta (FIGURA 7)

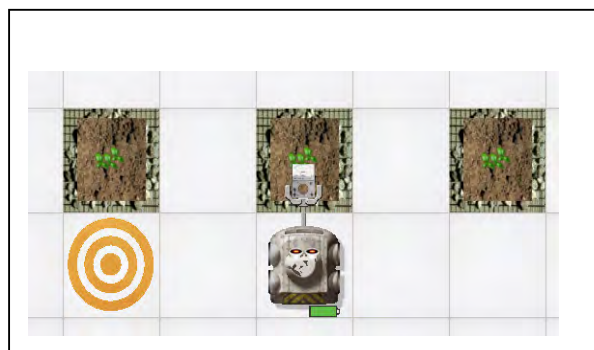


FIGURA 7 – Exemplo visual do exercício 7

8) Faça o robô calcular a média de água no grupo de plantas (FIGURA 8)

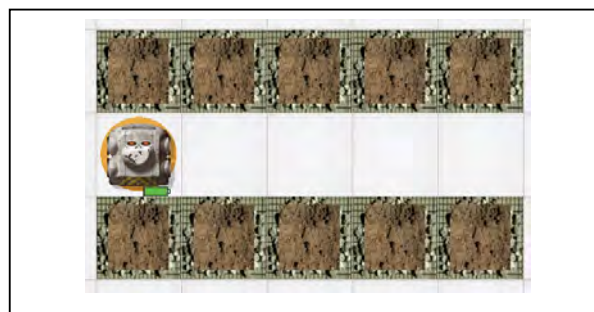


FIGURA 8 - Exemplo visual do exercício 8

9) Faça o robô calcular a média de água em cada planta após realizar 3 leituras. Estas leituras devem ser realizadas alternadamente entre uma planta e outra. (FIGURA 9)

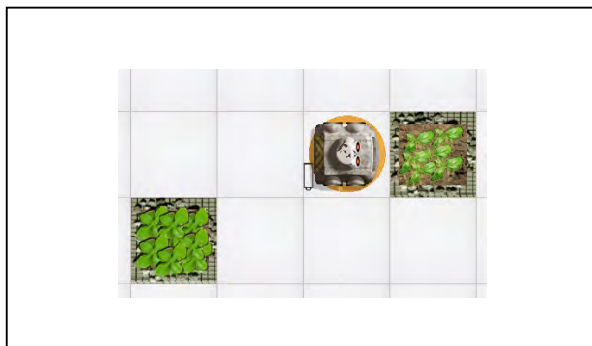


FIGURA 9 – Exemplo visual do exercício 9

10) Faça o robô verificar o quanto de água a planta está consumindo a cada 10 segundos (Taxa de consumo). Realizar a medição durante 1 minuto. (FIGURA 10)

11) Faça o robô verificar a saúde da planta a cada 10 segundos da seguinte forma: Taxa de saúde instantânea, ou seja, o valor em cada medição. Taxa de saúde média depois de 1 minuto. (FIGURA 10)

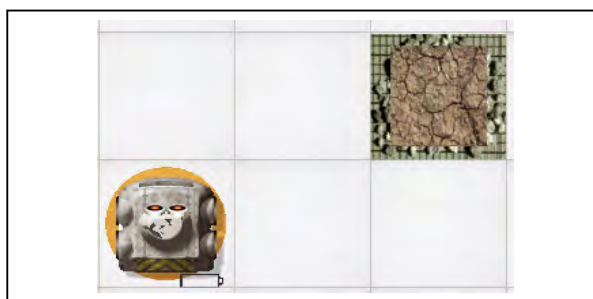


FIGURA 10 Exemplo visual do exercício 10 e 11

12) Enquanto o robô possuir baterias que o façam operar desenhe um programa que mantenha a planta saudável a maior parte do tempo. Imagine que a

planta para ser saudável precisa ser mantida na maior parte do tempo com:
(FIGURA 11.

- 50 unidades de água
- 50 unidades de fertilizante
- 0 quantidade de pestes

13) Crie uma seqüência de comandos que permita ao robô ficar vigiando a planta durante 1 minuto. Cada vez que ele detectar que a planta tem quantidade de água inferior a 40 unidades este deve irrigar para que o nível de água fique em 55 unidades. (FIGURA 11).



FIGURA 11 - Exemplo visual do exercício 12 e 13

14) Crie uma seqüência de comandos que permita ao robô ficar vigiando as duas plantas durante 2 minutos. Cada vez que o robô detectar que as duas plantas simultaneamente têm quantidade de água inferior a 40 unidades o robô deve irrigar ambas para que o nível de água em cada uma fique em 55 unidades. (FIGURA 12).

15) Crie uma seqüência de comandos que permita ao robô ficar vigiando as duas plantas durante 2 minutos. Cada vez que o robô detectar que pelo menos uma das duas plantas tem quantidade de água inferior a 40 unidades ambas as plantas serão irrigadas adicionando-se 20 unidades. (FIGURA 12).

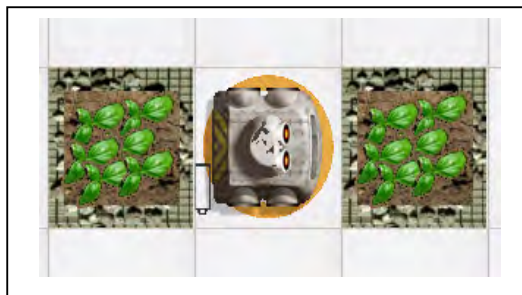


FIGURA 12 - Exemplo visual do exercício 14 e 15

16) Crie uma seqüência de comandos que permita ao robô avançar e consumir todas as baterias. As seguintes regras devem ser obedecidas: (FIGURA 13)

- Se a bateria for de 20 unidades você deve consumir a próxima
- Caso não exista mais baterias na frente do robô você deve virá-lo 180 graus e voltar.
- Se a bateria for maior que 20 unidades você deverá dormir por 5 segundos e depois continuar.



FIGURA 13 - Exemplo visual do exercício 16

Anexo 3 – Questionário Sociométrico

O questionário sociométrico proposto aos alunos continha uma única pergunta formulada da seguinte maneira: “Escolha 4 colegas que você deseja que façam parte do seu grupo para realizar trabalhos em equipe valendo nota. Entre estes 4 alunos você não poderá estar incluído”.

Uma visão gráfica das respostas ao questionário sociométrico:

O resultado do questionário é listado abaixo junto com informações adicionais como notas e média nas provas escritas individuais dentro do tema do currículo.

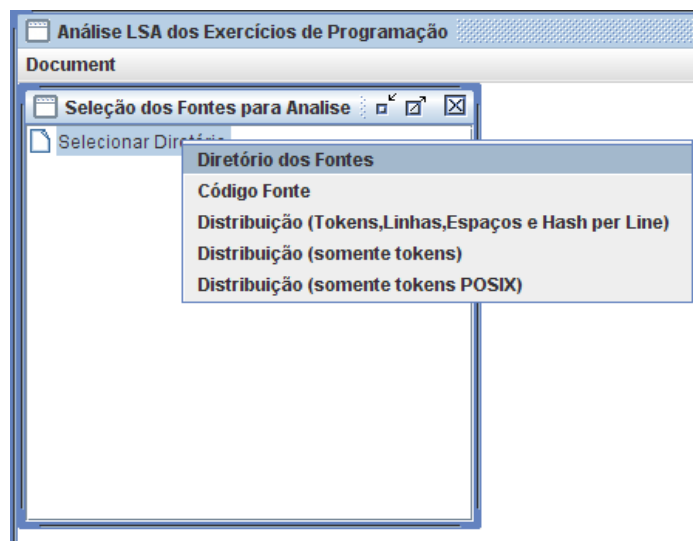


Na coluna onde se lê Colega 1 até 4 são os alunos apontados com seus respectivos códigos. Os alunos que não responderam ao questionário possuem as colunas dos respectivos colegas marcadas pelo símbolo “-”.

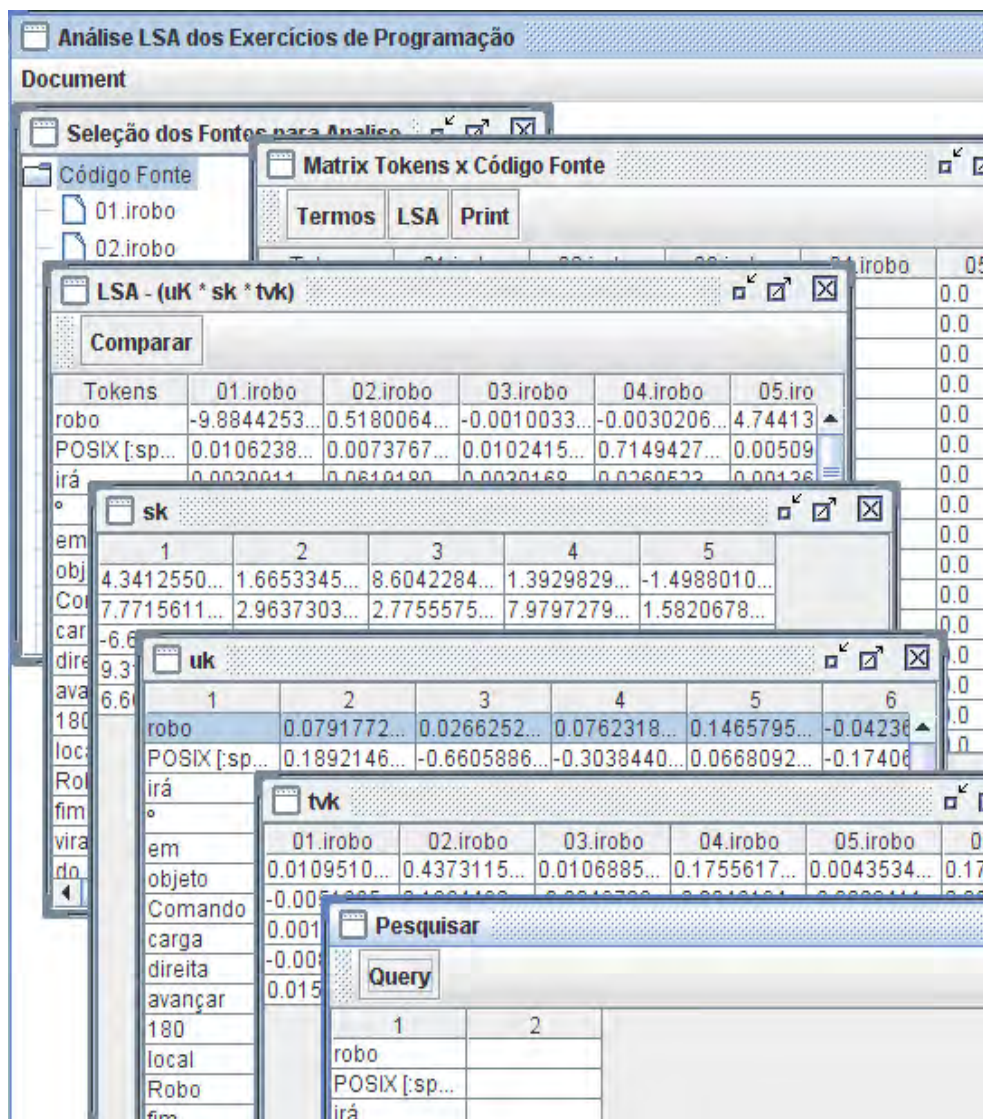
ID	Sexo	Média Provas	Colega1	Colega2	Colega3	Colega4
1	M	5,5	32	2	15	48
2	M	6,5	32	38	21	1
3	F	6,75	36	22	40	13
4	M	9,5	41	17	7	24
5	F	3,5	18	41	4	24
6	F	7,25	8	39	17	7
7	M	7,75	17	4	41	11
8	M	5	7	17	6	39
9	F	6	19	46	16	30
10	F	7,25	29	17	30	16
11	M	6,5	20	37	33	47
12	F	1,5	-	-	-	-
13	M	6,75	3	22	40	36
14	M	3,75	49	38	42	48
15	M	7	40	22	13	36
16	F	6,75	30	35	46	26
17	M	8,75	7	8	6	39
18	F	8	4	17	41	43
19	F	6	9	46	16	30
20	M	5	37	47	11	33
21	M	5,75	32	2	38	1
22	M	7	13	3	36	15
24	F	6	41	29	4	18
25	F	5,5	10	12	45	5
26	F	4,5	30	16	46	9
27	F	5	30	16	18	46
29	F	9	4	17	44	34
30	M	7,25	16	35	46	41
31	M	4,25	38	21	32	2
32	M	6	2	21	48	1
33	M	6,5	11	20	37	47
34	F	6,25	44	29	17	9
35	M	5,25	30	16	26	46
36	F	7,5	13	3	22	40
37	F	7,75	20	11	47	33
38	M	8	49	48	42	14
39	M	6	17	6	8	7
40	M	6,5	36	3	13	22
41	M	8	4	18	29	17
42	M	6,25	49	48	38	14
43	M	4,5	49	42	38	48
44	F	7,25	29	34	4	17
45	F	0,75	10	12	25	-
46	F	5,75	16	30	9	19
47	F	2,5	11	20	37	33
48	M	6,25	38	49	42	14
49	M	7,25	38	48	42	14

Anexo 4 – Software para calcular LSI

Para poder executar o cálculo LSI dos vários artefatos produzidos um software foi desenvolvido para esta finalidade. O software é capaz das seguintes funções:



- 1) Ler um diretório contendo o artefato de todos os alunos e executar o desmembramento do código com o objetivo de criar a matriz termo documento.
- 2) A matriz termo-documento pode ser ponderada usando a técnica da entropia
- 3) O procedimento de cálculo LSI da matriz termo-documento admite a configuração do parâmetro k para controlar a quantidade de valores singulares que serão utilizados na redução das dimensões da matriz
- 4) O ferramenta após o calculo é capaz de apresentar dos os componentes SVD gerados com a redução da matriz
- 5) O sistema também efetua a comparação entre todos os documentos entre si para gerar uma matriz de similaridade S_{ij} , onde s_{ij} representa a similaridade entre o artefato do aluno i com o aluno j .

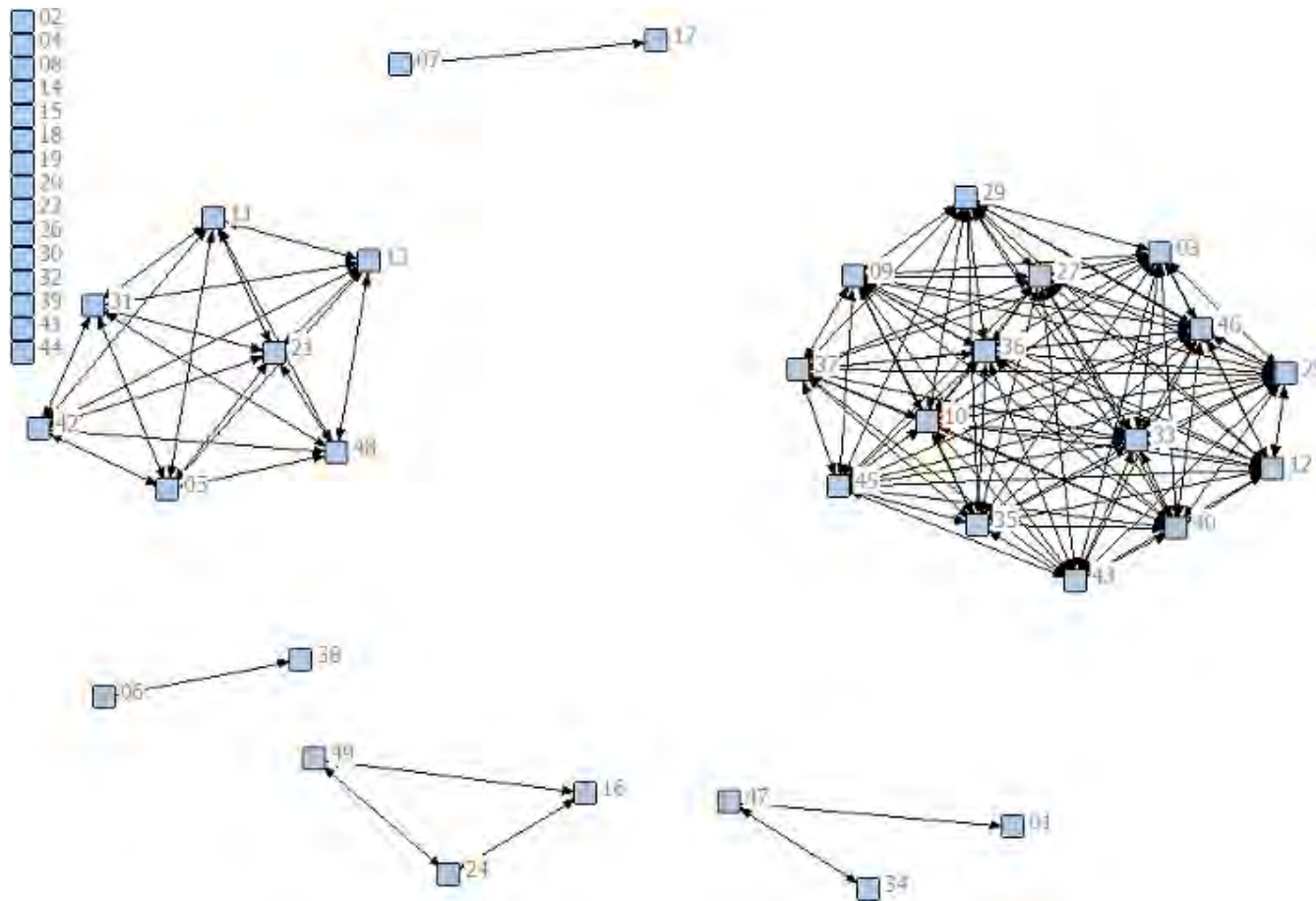


Apresentação da ferramenta desenvolvida para o cálculo das similaridades entre os trabalhos de programação.

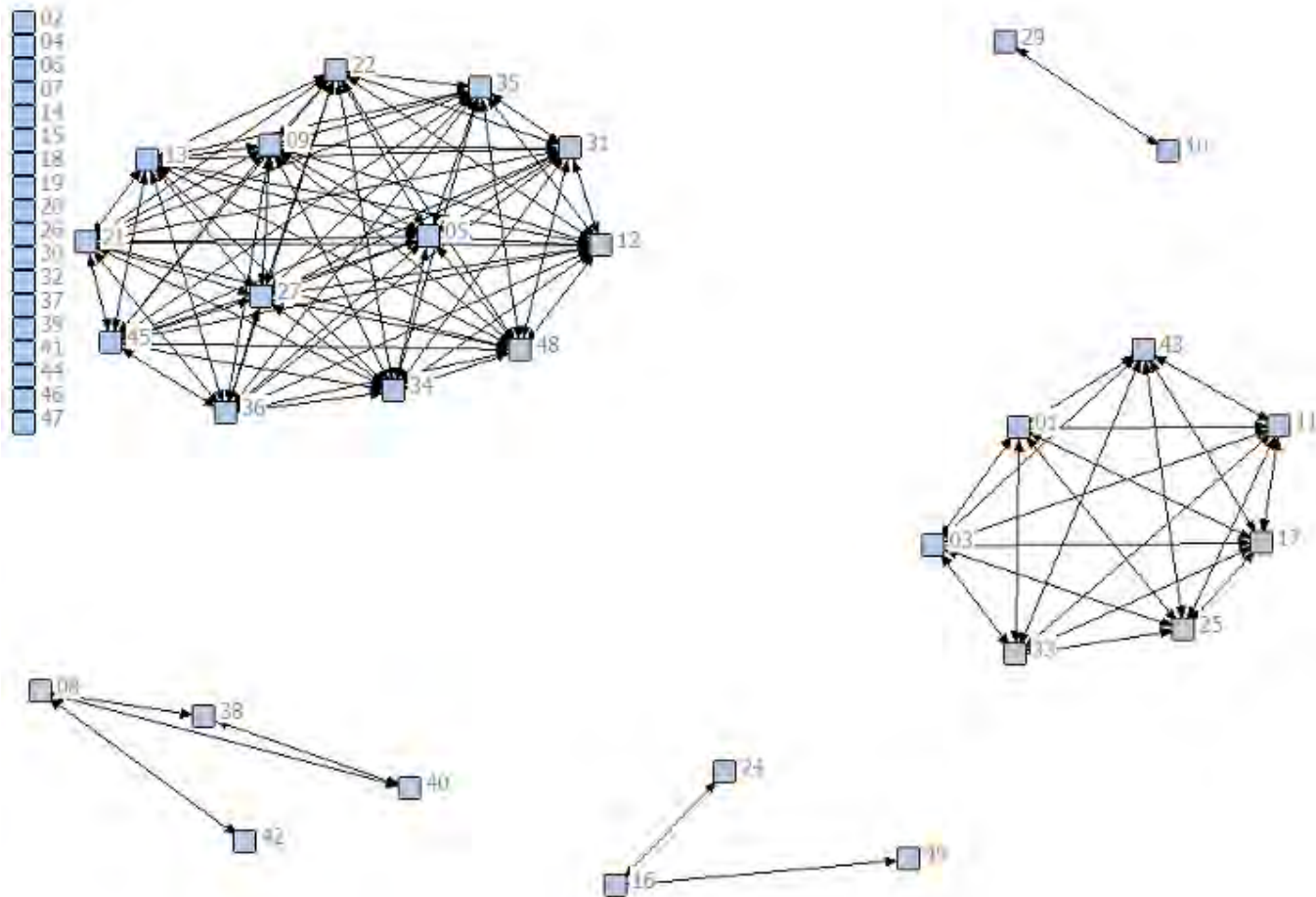
Anexo 5 – Dados de Similaridade

A Seguir, matriz de similaridade para cada artefato entregue pelos alunos.

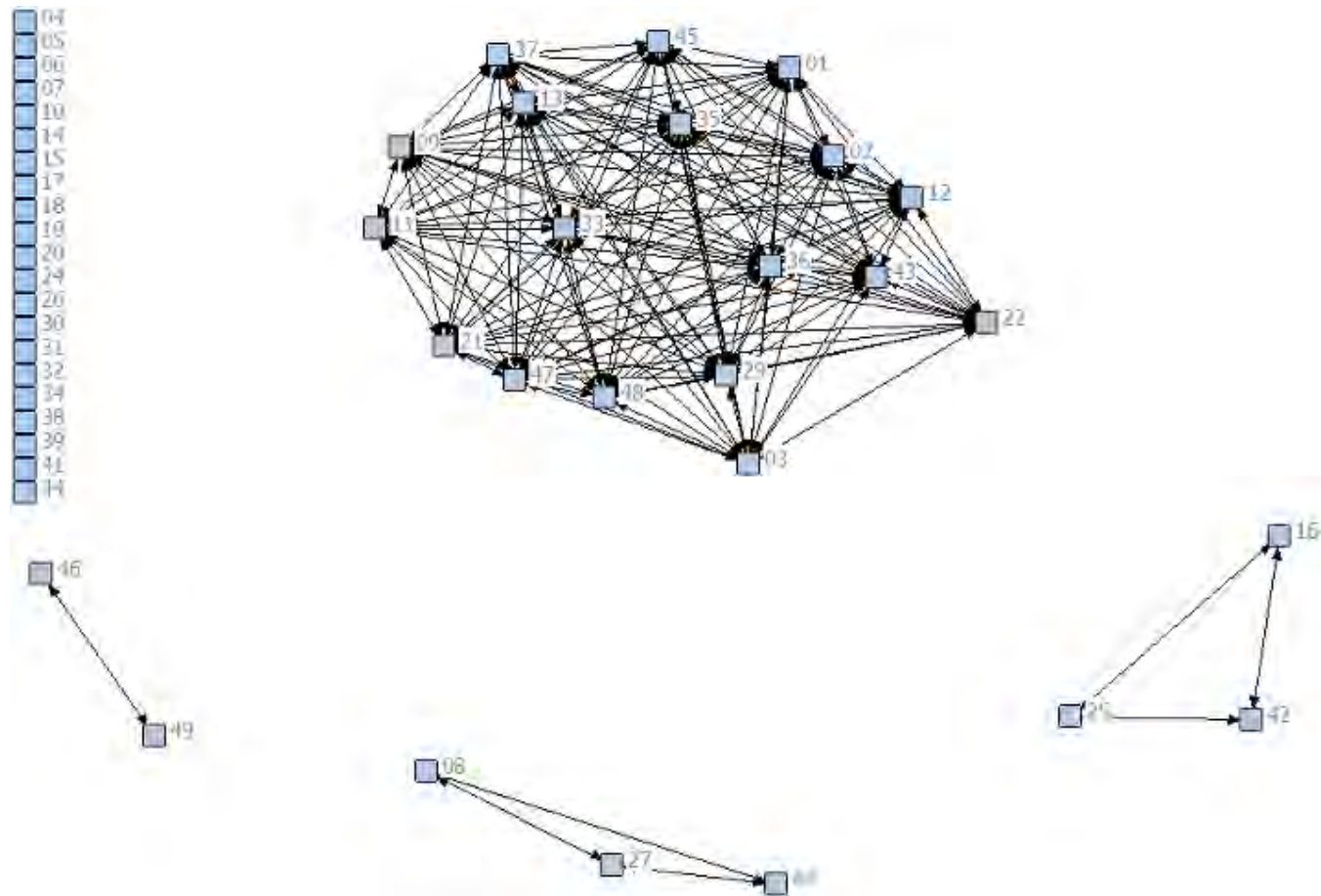
3) Rede desenhada utilizando-se os dados de similaridade gerados para o exercício 1 excluindo-se relações de similaridade menores que 0,9:



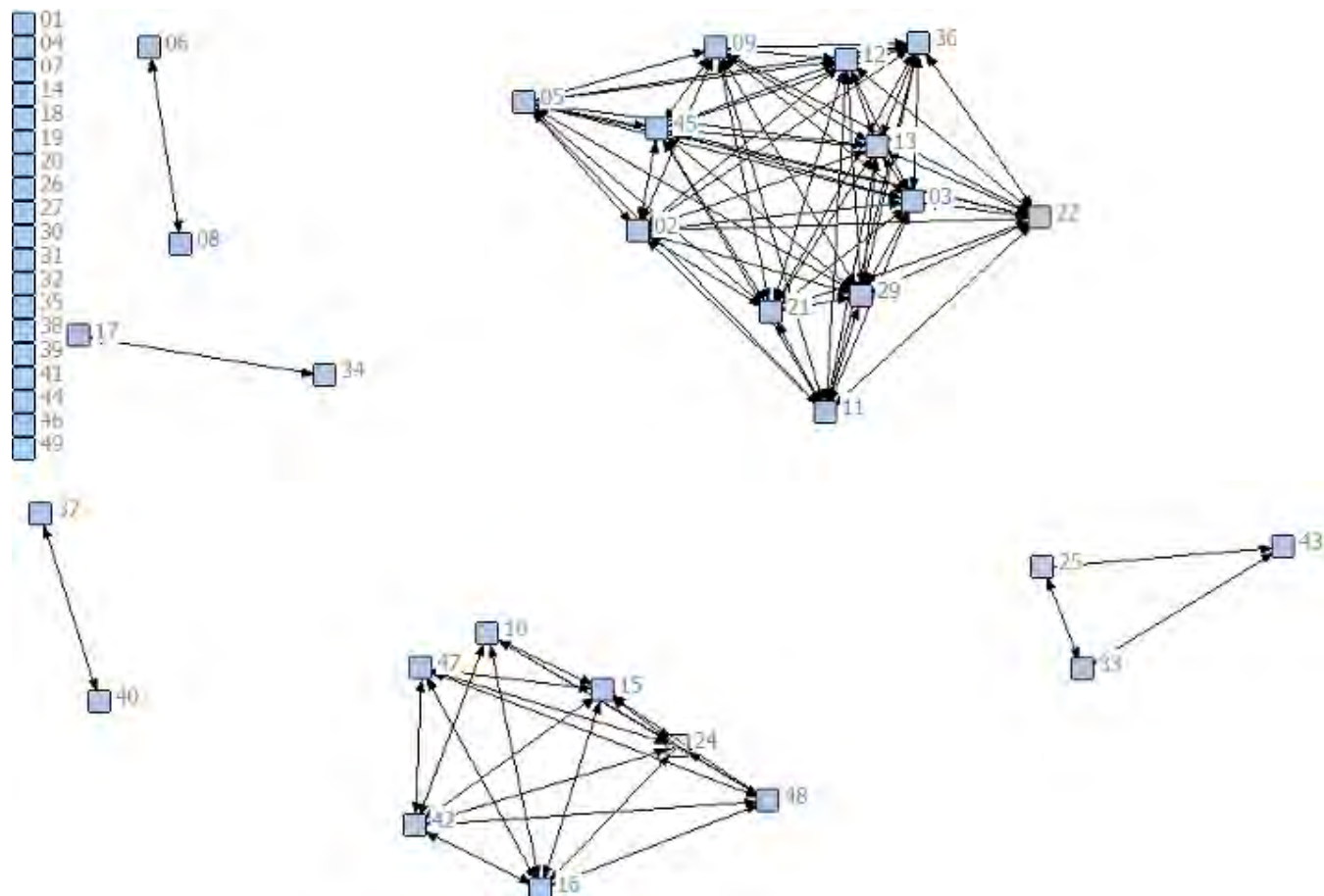
6) Rede desenhada utilizando-se os dados de similaridade gerados para o exercício 2 excluindo-se relações de similaridade menores que 0,9:



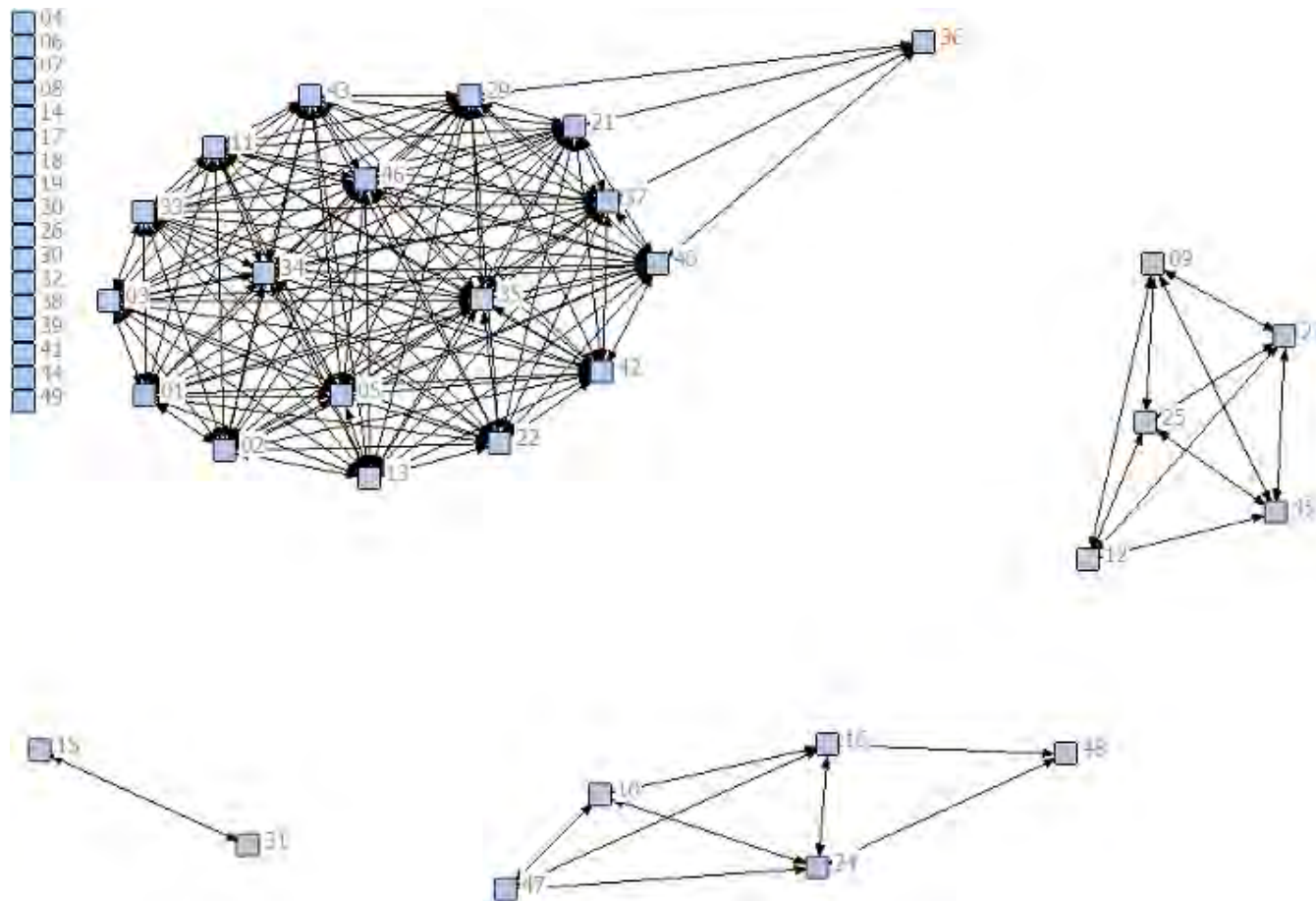
9) Rede desenhada utilizando-se os dados de similaridade gerados para o exercício 3 excluindo-se relações de similaridade menores que 0,9:



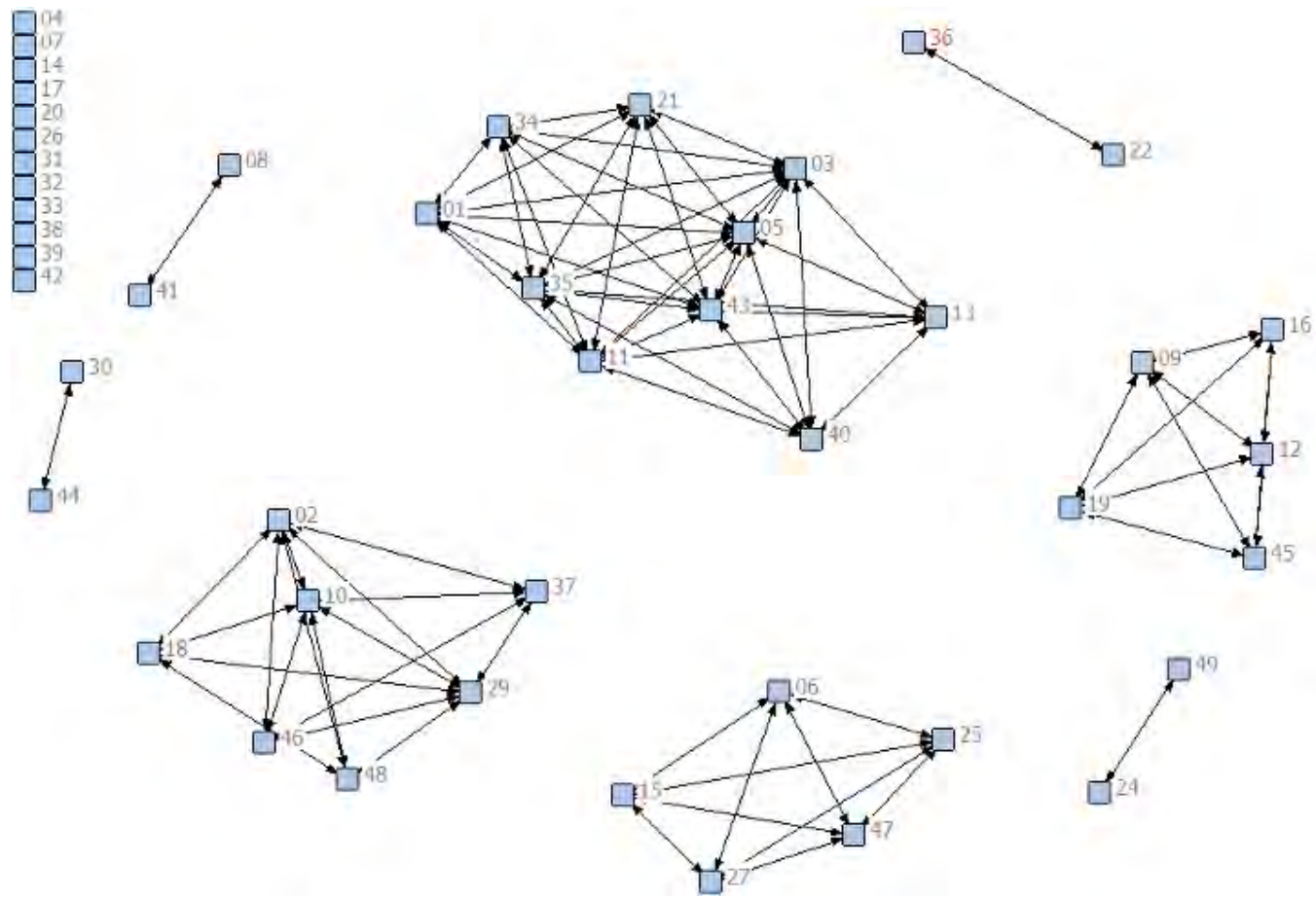
12) Rede desenhada utilizando-se os dados de similaridade gerados para o exercício 4 excluindo-se relações de similaridade menores que 0,9:



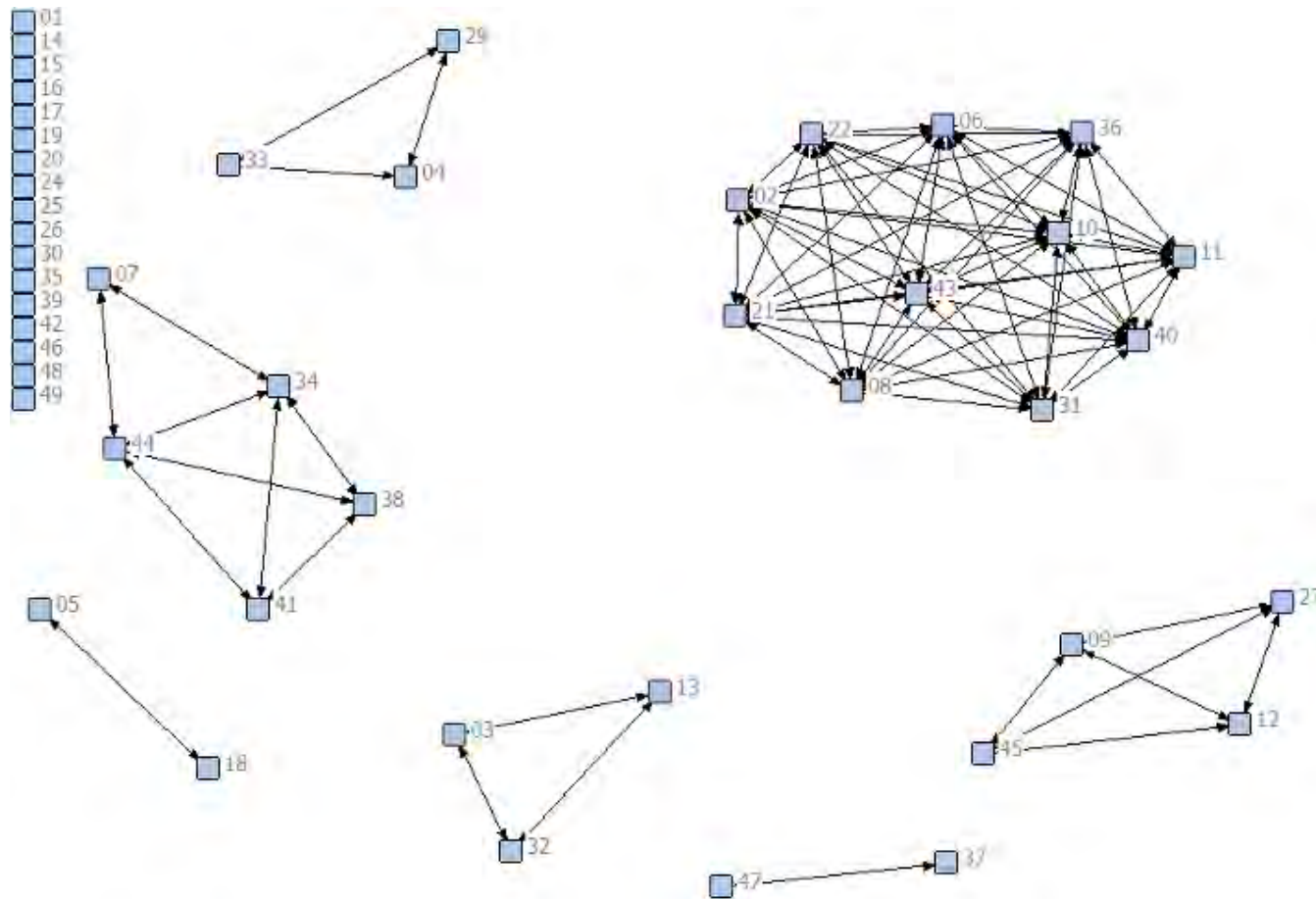
15) Rede desenhada utilizando-se os dados de similaridade gerados para o exercício 5 excluindo-se relações de similaridade menores que 0,9:



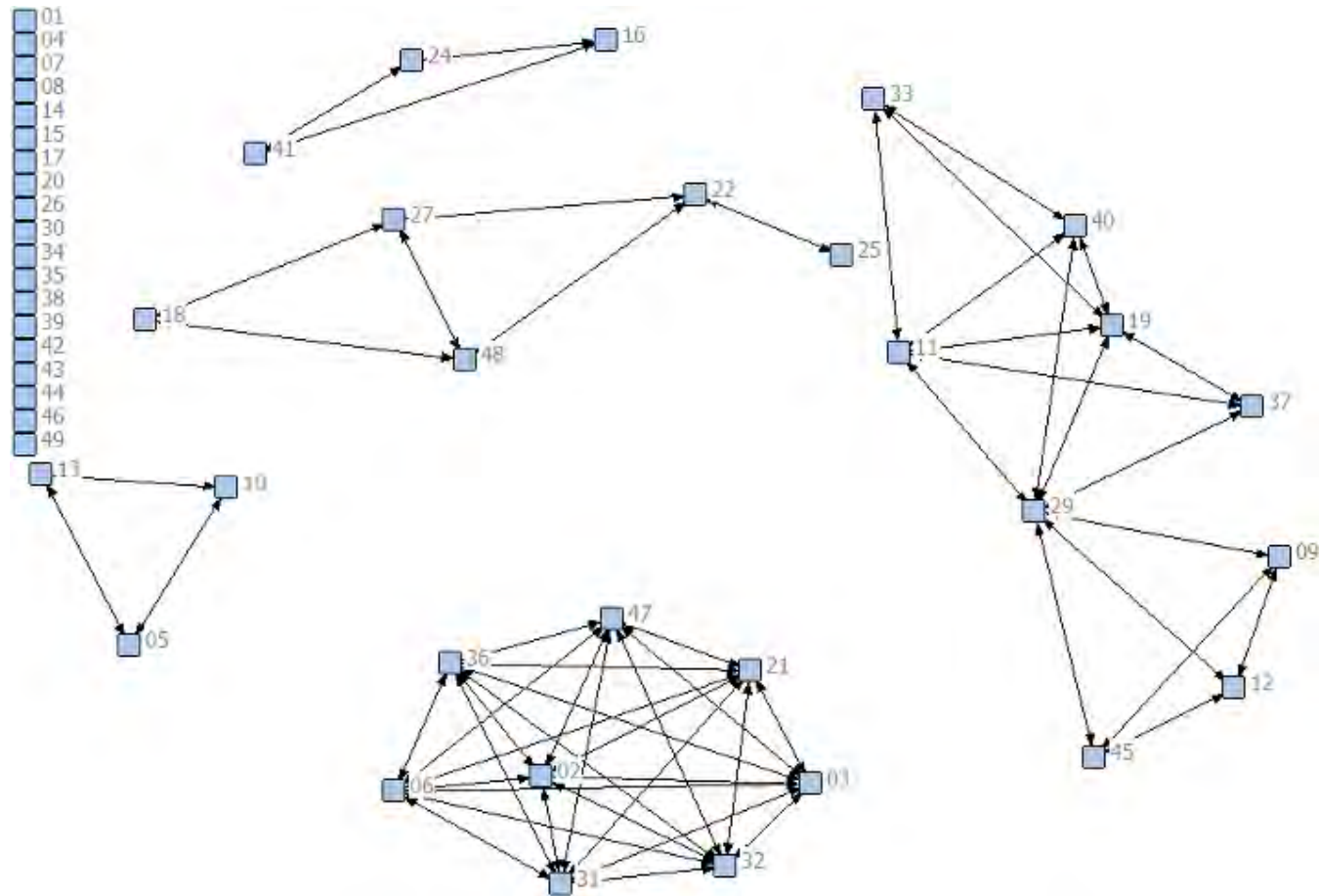
18) Rede desenhada utilizando-se os dados de similaridade gerados para o exercício 6 excluindo-se relações de similaridade menores que 0,9:



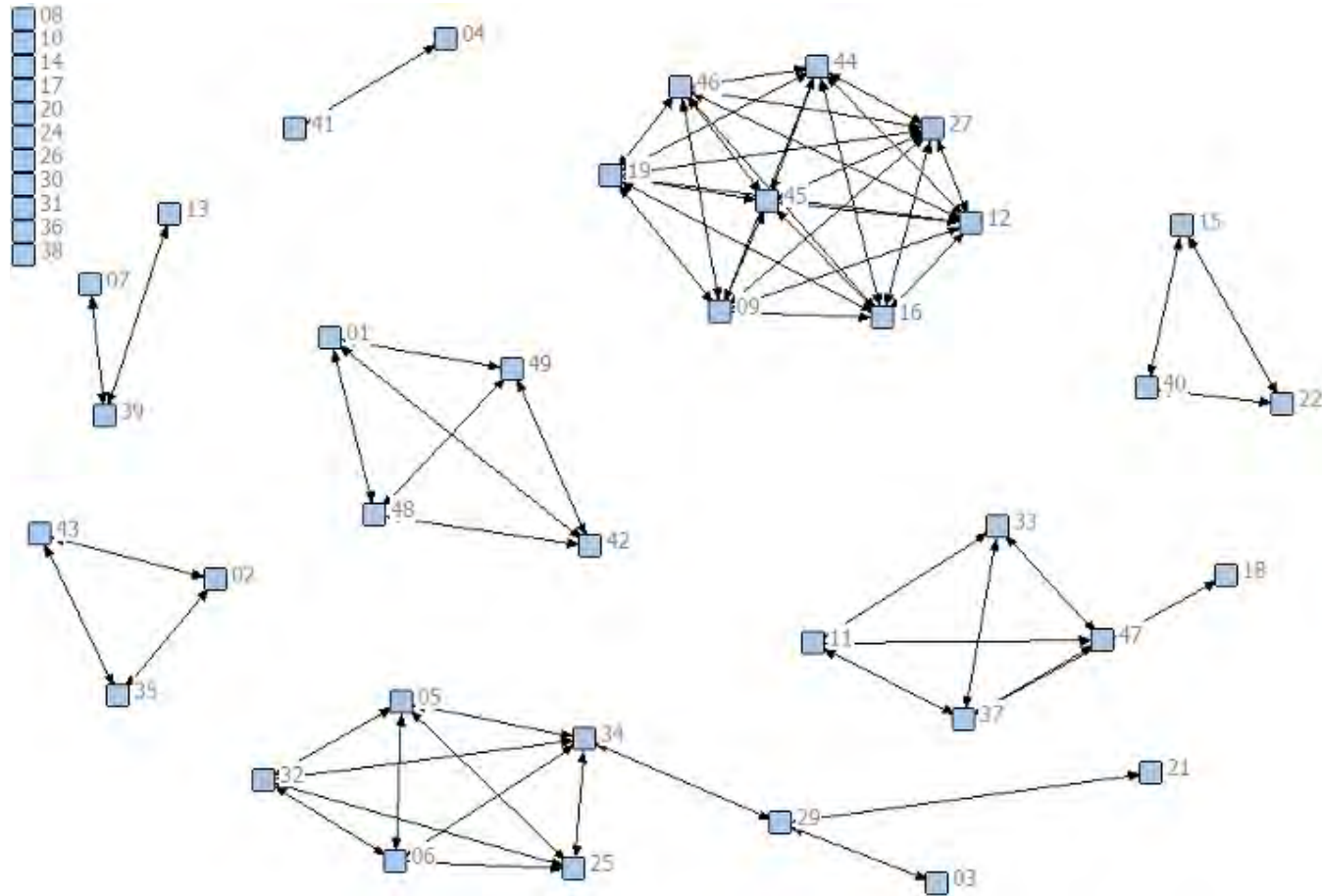
21) Rede desenhada utilizando-se os dados de similaridade gerados para o exercício 7 excluindo-se relações de similaridade menores que 0,9:



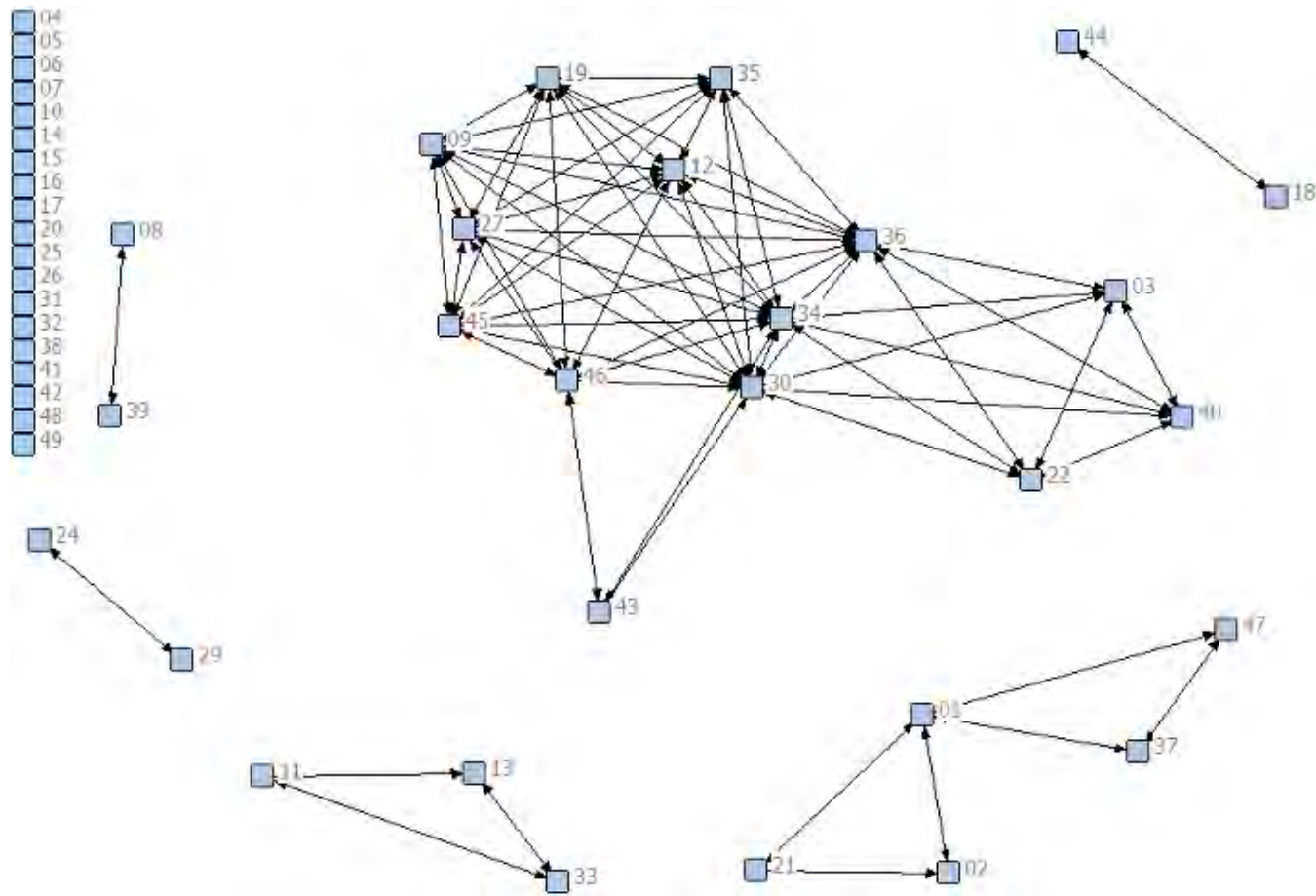
24) Rede desenhada utilizando-se os dados de similaridade gerados para o exercício 8 excluindo-se relações de similaridade menores que 0,9:



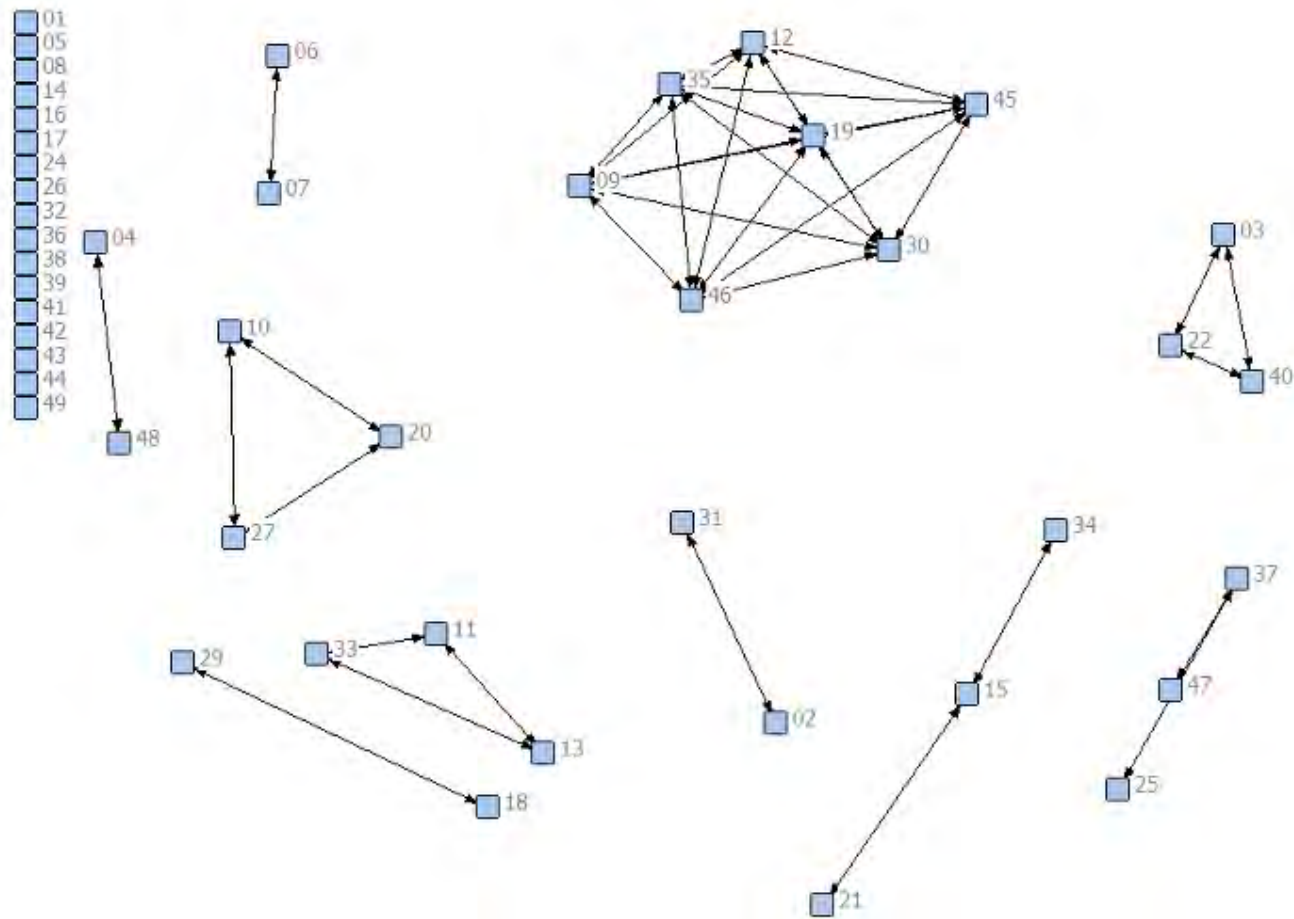
27) Rede desenhada utilizando-se os dados de similaridade gerados para o exercício 9 excluindo-se relações de similaridade menores que 0,9:



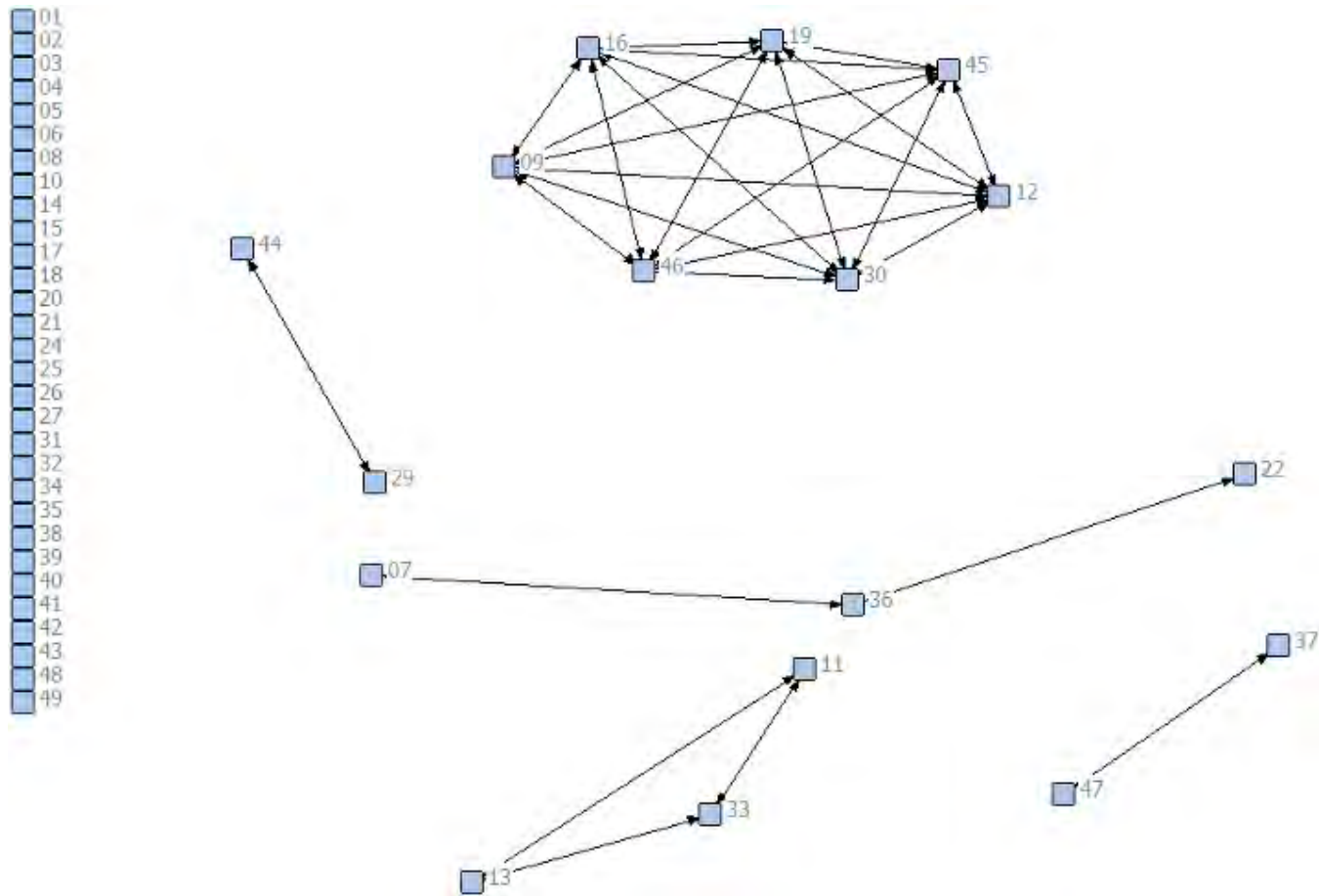
30) Rede desenhada utilizando-se os dados de similaridade gerados para o exercício 10 excluindo-se relações de similaridade menores que 0,9:



33) Rede desenhada utilizando-se os dados de similaridade gerados para o exercício 11 excluindo-se relações de similaridade menores que 0,9:



36) Rede desenhada utilizando-se os dados de similaridade gerados para o exercício 12 excluindo-se relações de similaridade menores que 0,9:



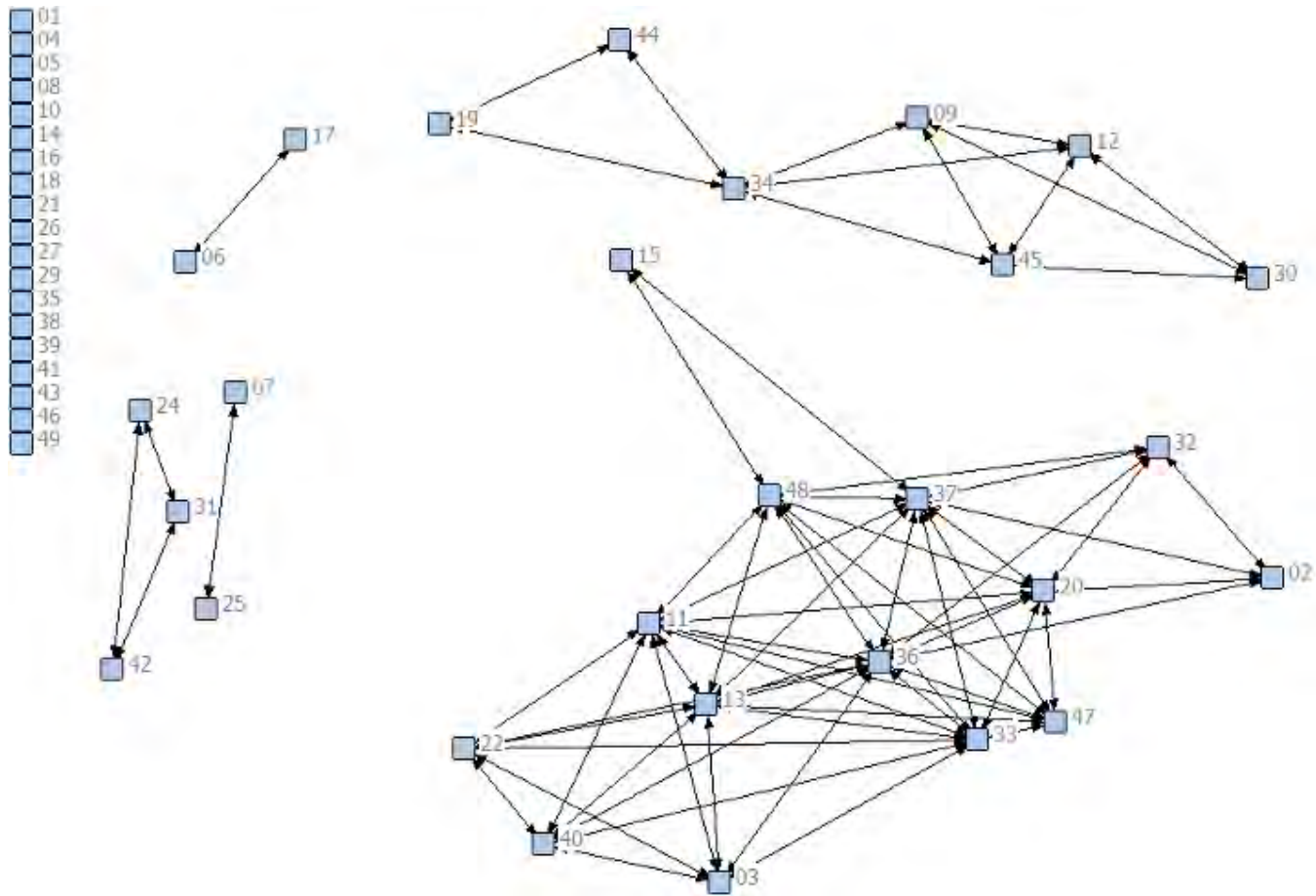
37) Matriz de similaridade para o Exercício 13 ajustado pelo fator de entropia e reduzido à k=10. Alunos de 1 até 24

Doc. x Doc.	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	24
01	1,00000	0,81593	0,84244	-0,00581	0,41525	0,14577	-0,01749	-0,06376	0,26442	0,00000	0,79977	0,26442	0,79977	0,00000	0,51289	0,19920	-0,00589	0,03457	0,62241	0,70394	0,42657	0,84244	0,30012
02	0,81593	1,00000	0,88472	0,00585	0,22410	0,06823	-0,09522	0,03023	0,49058	0,00000	0,81141	0,49058	0,81141	0,00000	0,77785	0,58113	-0,09499	-0,04240	0,62339	0,90336	0,47197	0,88472	0,04454
03	0,84244	0,88472	1,00000	-0,09577	0,32390	0,18920	0,04341	-0,02575	0,34477	0,00000	0,90173	0,34477	0,90173	0,00000	0,70127	0,37184	0,12544	0,07917	0,60980	0,89466	0,44313	1,00000	-0,03917
04	-0,00581	0,00585	-0,09577	1,00000	0,28544	-0,10206	0,00331	0,00825	-0,18486	0,00000	-0,04223	-0,01846	-0,04223	0,00000	0,13573	-0,00408	0,00372	0,03869	0,09384	-0,03951	-0,05234	-0,09577	-0,10444
05	0,41525	0,22410	0,32390	0,28544	1,00000	0,48175	0,32658	-0,17343	0,21902	0,00000	0,28627	0,21902	0,28627	0,00000	0,01795	-0,13903	0,58988	0,79800	0,33018	0,18022	0,36778	0,32390	0,03337
06	0,14577	0,06823	0,18920	-0,01026	0,48175	1,00000	-0,14570	0,11545	-0,00372	0,00000	0,35398	-0,00372	0,35398	0,00000	0,06510	-0,02432	0,92923	0,26766	0,00845	0,17500	-0,03157	0,18920	0,24377
07	-0,01749	-0,09522	0,04341	0,00331	0,32658	-0,14570	1,00000	0,01542	-0,00803	0,00000	-0,06916	-0,00803	-0,06916	0,00000	0,08920	0,06096	-0,06553	0,64430	0,10149	0,00120	-0,00511	0,04341	0,08078
08	-0,06376	0,03023	-0,02575	0,00825	-0,17343	0,11545	0,01542	1,00000	-0,05435	0,00000	0,01828	-0,05435	0,01828	0,00000	0,34781	0,06500	-0,00054	-0,13425	0,23242	0,03829	0,05274	-0,02576	0,10021
09	0,26442	0,49058	0,34477	-0,01846	0,21902	-0,00372	-0,00803	-0,05435	1,00000	0,00000	0,00303	1,00000	0,00303	0,00000	0,08252	0,00513	-0,03287	0,05224	0,82535	0,18618	-0,20693	0,34477	0,05345
10	0,00000	0,00000	0,00000	0,00000	0,00000	0,00000	0,00000	0,00000	0,00000	1,00000	0,00000	0,00000	0,00000	0,00000	0,00000	0,00000	0,00000	0,00000	0,00000	0,00000	0,00000	0,00000	0,00000
11	0,79977	0,81141	0,90173	-0,04223	0,28527	0,35398	-0,06916	0,01828	0,00303	0,00000	1,00000	0,00303	1,00000	0,00000	0,77308	0,52330	0,22776	0,03234	0,28175	0,93057	0,56796	0,90173	0,01513
12	0,26442	0,49058	0,34477	-0,01846	0,21902	-0,00372	-0,00803	-0,05435	1,00000	0,00000	0,00303	1,00000	0,00303	0,00000	0,08252	0,00513	-0,03287	0,05224	0,82535	0,18618	-0,20693	0,34477	0,05345
13	0,79977	0,81141	0,90173	-0,04223	0,28527	0,35398	-0,06916	0,01828	0,00303	0,00000	1,00000	0,00303	1,00000	0,00000	0,77308	0,52330	0,22776	0,03234	0,28175	0,93057	0,56796	0,90173	0,01513
14	0,00000	0,00000	0,00000	0,00000	0,00000	0,00000	0,00000	0,00000	0,00000	0,00000	0,00000	0,00000	0,00000	0,00000	0,00000	0,00000	0,00000	0,00000	0,00000	0,00000	0,00000	0,00000	0,00000
15	0,51289	0,77785	0,70127	0,13573	0,01795	0,06510	0,09820	0,34781	0,08252	0,00000	0,77308	0,08252	0,77308	0,00000	1,00000	0,81532	-0,11833	-0,09150	0,31361	0,89999	0,51498	0,70127	-0,16146
16	0,19920	0,58113	0,37184	-0,00408	-0,13903	-0,02432	0,06096	0,06500	0,00513	0,00000	0,35398	-0,00513	0,35398	0,00000	0,01532	0,00000	-0,12338	-0,08806	-0,07618	0,73456	0,47572	0,37184	-0,20133
17	-0,00589	-0,09499	0,12544	0,00372	0,58988	0,92923	-0,06553	-0,00054	-0,03287	0,00000	0,22776	-0,03287	0,22776	0,00000	-0,11833	-0,21238	1,00000	0,43578	-0,02573	0,03824	-0,03340	0,12544	0,04232
18	0,03457	-0,04240	0,07917	0,03869	0,79800	0,26766	0,64430	-0,13425	0,05224	0,00000	0,03234	0,05224	0,03234	0,00000	-0,09150	-0,08806	0,43578	1,00000	0,06496	-0,01095	0,32316	0,07917	-0,01756
19	0,62241	0,62339	0,60980	0,09384	0,33018	0,00845	0,10149	-0,23242	0,82535	0,00000	0,28175	0,82535	0,28175	0,00000	0,31361	-0,07618	-0,02573	0,06496	1,00000	0,36074	-0,02510	0,60980	0,08860
20	0,70394	0,90336	0,89466	-0,03951	0,16022	0,17500	0,00120	0,03829	0,18618	0,00000	0,93057	0,18618	0,93057	0,00000	0,89999	0,73456	0,03824	-0,01095	0,36074	1,00000	0,56424	0,89466	-0,10942
21	0,42657	0,47197	0,44313	-0,05234	0,36778	-0,03157	0,05274	-0,20693	0,06796	0,00000	0,51498	-0,20693	0,06796	0,00000	0,51498	0,47572	-0,03340	0,32316	0,28175	0,93057	0,56796	0,90173	0,01513
22	0,84244	0,88472	1,00000	-0,09577	0,32390	0,18920	0,04341	-0,02575	0,34477	0,00000	0,90173	0,34477	0,90173	0,00000	0,70127	0,37184	0,12544	0,07917	0,60980	0,89466	0,44313	1,00000	-0,03917
24	0,30012	0,04454	-0,03917	-0,10444	0,03337	0,24377	0,08078	0,10021	0,05345	0,00000	0,01513	0,05345	0,01513	0,00000	-0,16146	-0,20133	0,04232	-0,01756	0,08860	-0,10942	-0,39412	-0,03917	1,00000
25	0,12387	0,09159	0,19456	-0,17733	0,45283	-0,05331	0,93829	-0,04092	0,19317	0,00000	0,04368	0,19317	0,04368	0,00000	0,14088	0,13975	0,00680	0,73379	0,23282	0,13363	0,13704	0,19456	0,07552
26	0,00000	0,00000	0,00000	0,00000	0,00000	0,00000	0,00000	0,00000	0,00000	0,00000	0,00000	0,00000	0,00000	0,00000	0,00000	0,00000	0,00000	0,00000	0,00000	0,00000	0,00000	0,00000	0,00000
27	0,00000	0,00000	0,00000	0,00000	0,00000	0,00000	0,00000	0,00000	0,00000	0,00000	0,00000	0,00000	0,00000	0,00000	0,00000	0,00000	0,00000	0,00000	0,00000	0,00000	0,00000	0,00000	0,00000
29	0,70479	0,70590	0,87943	-0,07707	0,50360	0,26220	0,02373	-0,10381	0,10389	0,00000	0,85853	0,10389	0,85853	0,00000	0,54945	0,27108	0,31044	0,35220	0,36612	0,77981	0,67316	0,87943	-0,19739
30	0,03976	0,32516	0,22988	0,04357	0,09566	0,07309	-0,01957	-0,06891	0,92943	0,00000	-0,09665	0,92943	-0,09665	0,00000	-0,00149	-0,05957	0,09264	0,01408	0,70709	0,08791	-0,43642	0,22988	0,04101
31	0,35335	0,08648	-0,00004	-0,06708	0,02758	0,28290	0,02863	0,04702	0,05089	0,00000	0,07503	0,05089	0,07503	0,00000	-0,11615	-0,14896	0,05103	-0,09977	0,06130	-0,05403	-0,40039	-0,00004	0,98523
32	0,72348	0,98202	0,87164	-0,01437	0,18595	0,00278	-0,08815	0,00203	0,49047	0,00000	0,78340	0,49047	0,78340	0,00000	0,79506	0,63529	-0,11883	-0,02173	0,58698	0,91429	0,52467	0,87164	-0,12070
33	0,79977	0,81141	0,90173	-0,04223	0,28527	0,35398	-0,06916	0,01828	0,00303	0,00000	1,00000	0,00303	1,00000	0,00000	0,77308	0,52330	0,22776	0,03234	0,28175	0,93057	0,56796	0,90173	0,01513
34	0,49137	0,62749	0,58223	-0,00291	0,26750	0,00340	-0,04066	-0,01286	0,95122	0,00000	0,20711	0,95122	0,20711	0,00000	0,18842	-0,04245	-0,02850	0,01237	0,93439	0,32638	-0,10653	0,58223	0,08408
35	-0,46658	-0,11024	-0,39771	-0,02519	-0,05688	0,16874	0,00201	0,03167	0,12882	0,00000	-0,27905	0,12882	-0,27905	0,00000	0,06000	0,51169	0,06238	0,09049	-0,29669	-0,07066	0,01237	-0,39771	-0,14575
36	0,74036	0,90273	0,90139	-0,02792	0,17098	0,18229	-0,00982	0,03330	0,16369	0,00000	0,94561	0,16369	0,94561	0,00000	0,89548	0,71389	0,03226	-0,03271	0,36091	0,99710	0,56699	0,90139	-0,08735
37	0,70297	0,90284	0,89402	-0,03929	0,15858	0,17373	0,00048	0,03513	0,18472	0,00000	0,93038	0,18472	0,93038	0,00000	0,90032	0,73534	0,03409	-0,01213	0,35922	1,00000	0,55470	0,89402	-0,11065
38	-0,18111	0,12578	0,21374	-0,00260	-0,45752	0,01798	0,06150	0,00057	-0,10778	0,00000	0,25181	-0,10778	0,25181	0,00000	0,37960	0,38169	0,03577	-0,22495	-0,13161	0,36031	-0,26680	0,21374	-0,09010
39	0,00000	0,00000	0,00000	0,00000	0,00000	0,00000	0,00000	0,00000	0,00000	0,00000	0,00000	0,00000	0,00000	0,00000	0,00000	0,00000	0,00000	0,00000	0,00000	0,00000	0,00000	0,00000	0,00000
40	0,84244	0,88472	1,00000	-0,09577	0,32390	0,18920	0,04341	-0,02575	0,34477	0,00000	0,90173	0,34477	0,90173	0,00000	0,70127	0,37184	0,12544	0,07917	0,60980	0,89466	0,44313	1,00000	-0,03917
41	-0,05361	0,07451	-0,11850	0,03352	0,07351	-0,09377	-0,06026	-0,01814	-0,02186														

38) Matriz de similaridade para o Exercício 13 ajustado pelo fator de entropia e reduzido à k=10. Alunos de 25 até 49

25	26	27	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49
0.12387	0.00000	0.00000	0.70479	0.03976	0.35335	0.72348	0.79977	0.49137	-0.46658	0.74036	0.70297	-0.18111	0.00000	0.84244	-0.05361	0.35305	0.53964	0.57123	0.26442	0.24786	0.55479	0.69159	0.41385
0.09159	0.00000	0.00000	0.70950	0.32516	0.08648	0.98202	0.81141	0.62749	-0.11024	0.90273	0.90284	0.12578	0.00000	0.88472	0.07451	0.08614	0.18696	0.63105	0.49058	0.65820	0.74282	0.89084	0.32630
0.19456	0.00000	0.00000	0.67943	0.22988	-0.00004	0.87164	0.90173	0.55223	-0.39771	0.90139	0.89402	0.21374	0.00000	1.00000	-0.11850	-0.00041	0.20601	0.58787	0.34477	0.39786	0.80221	0.85679	0.18154
-0.17733	0.00000	0.00000	-0.07707	0.04357	-0.06708	-0.01437	-0.04223	-0.00291	-0.02519	-0.02792	-0.03929	-0.00260	0.00000	-0.09577	0.03352	-0.06705	0.39673	0.09979	-0.01846	0.02021	-0.05969	-0.02252	0.08073
0.45283	0.00000	0.00000	0.60360	0.09566	0.02758	0.18595	0.28527	0.26750	-0.05688	0.17098	0.15858	-0.45752	0.00000	0.32390	0.07351	0.02730	0.67302	0.31718	0.21902	0.04837	0.15035	0.10479	-0.02474
-0.05331	0.00000	0.00000	0.26220	0.07309	0.28290	0.00278	0.35398	0.00340	0.16874	0.18229	0.17373	0.01798	0.00000	0.18920	-0.09377	0.28270	0.56868	-0.02085	-0.00372	-0.02856	0.28815	0.09006	0.19241
0.93829	0.00000	0.00000	0.02373	-0.01987	0.02663	-0.08815	-0.06916	-0.04066	0.00201	-0.00982	0.00048	0.06150	0.00000	0.04341	-0.06026	0.02653	0.10402	0.04814	-0.00603	-0.06778	-0.07362	-0.03315	0.01000
-0.04092	0.00000	0.00000	-0.10381	-0.06891	0.04702	0.00203	0.01828	-0.01286	0.03167	0.03330	0.03513	0.00067	0.00000	-0.02576	-0.01814	0.04699	0.02247	0.14362	-0.06435	0.00293	0.01448	0.00186	-0.09842
0.19317	0.00000	0.00000	0.10389	0.92943	0.05089	0.49047	0.00303	0.95122	0.12882	0.16369	0.18472	-0.10778	0.00000	0.34477	-0.02186	0.05063	0.06869	0.86704	1.00000	0.87861	-0.05809	0.12310	-0.22104
0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
0.04368	0.00000	0.00000	0.85853	-0.09565	0.07503	0.78340	1.00000	0.20711	-0.27905	0.94561	0.93038	0.25181	0.00000	0.90173	-0.05783	0.07473	0.28972	0.25322	0.00303	0.18036	0.92884	0.91533	0.43353
0.19317	0.00000	0.00000	0.10389	0.92943	0.05089	0.49047	0.00303	0.95122	0.12882	0.16369	0.18472	-0.10778	0.00000	0.34477	-0.02186	0.05063	0.06869	0.86704	1.00000	0.87861	-0.05809	0.12310	-0.22104
0.04368	0.00000	0.00000	0.85853	-0.09565	0.07503	0.78340	1.00000	0.20711	-0.27905	0.94561	0.93038	0.25181	0.00000	0.90173	-0.05783	0.07473	0.28972	0.25322	0.00303	0.18036	0.92884	0.91533	0.43353
0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
0.14088	0.00000	0.00000	0.58495	-0.01149	-0.11615	0.79806	0.77308	0.18842	0.06000	0.89848	0.90032	0.37960	0.00000	0.70127	-0.09375	-0.11640	0.01941	0.26903	0.08252	0.41830	0.83668	0.90214	0.43784
0.13975	0.00000	0.00000	0.27108	-0.08987	-0.14896	0.63829	0.22330	-0.04245	0.51169	0.71389	0.73534	0.38169	0.00000	0.37184	0.03935	-0.14910	-0.20314	-0.07607	0.00513	0.46203	0.70925	0.77087	0.57507
0.00680	0.00000	0.00000	0.31044	0.09264	0.05103	-0.11983	0.22776	-0.02890	0.06238	0.03226	0.03409	0.03577	0.00000	0.12544	-0.05928	0.05087	0.43312	-0.06513	-0.03287	-0.14843	0.19905	-0.05556	-0.11227
0.73379	0.00000	0.00000	0.35220	0.01408	-0.09977	-0.02173	0.03234	0.01237	0.09049	-0.03271	-0.01213	-0.22495	0.00000	0.07917	0.32549	-0.09995	0.28653	0.02316	-0.08224	-0.07843	0.03166	-0.06230	-0.24102
0.23282	0.00000	0.00000	0.36612	0.70709	0.06130	0.86698	0.28175	0.93439	-0.29669	0.36091	0.35922	-0.13161	0.00000	0.60980	-0.13580	0.08099	0.26729	0.99139	0.82535	0.65244	0.11675	0.29002	-0.21062
0.13363	0.00000	0.00000	0.58495	-0.01149	-0.11615	0.79806	0.77308	0.18842	0.06000	0.89848	0.90032	0.37960	0.00000	0.70127	-0.09375	-0.11640	0.01941	0.26903	0.08252	0.41830	0.83668	0.90214	0.43784
0.13704	0.00000	0.00000	0.67316	-0.43642	-0.40039	0.82467	0.67996	-0.10653	0.01237	0.55699	0.55470	-0.25680	0.00000	0.44513	-0.18428	-0.40056	-0.02870	-0.04540	-0.20693	-0.01900	0.62884	0.59920	0.13805
0.19456	0.00000	0.00000	0.67943	0.22988	-0.00004	0.87164	0.90173	0.55223	-0.39771	0.90139	0.89402	0.21374	0.00000	1.00000	-0.11850	-0.00041	0.20601	0.58787	0.34477	0.39786	0.80221	0.85679	0.18154
0.07552	0.00000	0.00000	-0.19739	0.04101	0.88523	-0.12070	0.01513	0.08408	-0.14575	-0.08735	-0.11065	-0.09010	0.00000	-0.03917	0.30960	0.98524	0.59774	0.09786	0.05345	-0.05235	-0.24232	-0.14036	0.38347
1.00000	0.00000	0.00000	0.17872	0.10969	0.02086	0.10357	0.04368	0.14873	0.09738	0.11737	0.13250	-0.05618	0.00000	0.19456	-0.01526	0.02067	0.11474	0.19487	0.19317	0.12964	0.03198	0.08786	0.01039
0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
0.17872	0.00000	0.00000	1.00000	0.00577	-0.19582	0.72943	0.85853	0.30582	-0.39837	0.77859	0.77944	0.12343	0.00000	0.87943	0.11142	-0.19594	0.16472	0.32993	0.10389	0.13502	0.81187	0.75192	-0.01298
0.10969	0.00000	0.00000	0.00577	1.00000	0.02934	0.34081	-0.09565	0.85197	-0.13691	0.05263	0.08668	0.17025	0.00000	0.22988	0.01381	0.02915	-0.02378	0.73482	0.92943	0.80967	-0.10232	0.01060	-0.33663
0.02086	0.00000	0.00000	-0.19582	0.02934	1.00000	-0.08889	0.07503	0.08726	-0.12738	-0.02452	-0.05523	-0.09003	0.00000	-0.00004	0.19415	1.00000	0.64456	0.10099	0.05089	-0.02905	-0.19081	-0.08045	0.50087
0.10357	0.00000	0.00000	0.72943	0.34081	-0.08889	1.00000	0.78340	0.60710	-0.05404	0.90391	0.91402	0.18577	0.00000	0.87164	0.07722	-0.08622	0.03064	0.59265	0.49047	0.69810	0.77757	0.90631	0.23451
0.04368	0.00000	0.00000	0.85853	-0.09565	0.07503	0.78340	1.00000	0.20711	-0.27905	0.94561	0.93038	0.25181	0.00000	0.90173	-0.05783	0.07473	0.28972	0.25322	0.00303	0.18036	0.92884	0.91533	0.43353
0.14873	0.00000	0.00000	0.30582	0.85197	0.08726	0.60710	0.20711	1.00000	-0.14412	0.31659	0.32490	-0.10416	0.00000	0.55223	-0.05835	0.08696	0.16443	0.96300	0.95122	0.79603	0.07873	0.26210	-0.20804
0.09738	0.00000	0.00000	-0.39837	0.13691	-0.12738	-0.05404	-0.27905	-0.14412	1.00000	-0.10398	-0.07058	-0.02873	0.00000	-0.39771	0.06111	-0.12737	-0.18290	-0.26378	0.12882	0.39656	-0.05412	-0.06156	0.25111
0.11737	0.00000	0.00000	0.77859	0.08263	-0.02452	0.90391	0.94561	0.31659	-0.10398	1.00000	0.99707	0.32295	0.00000	0.90139	-0.08167	-0.02484	0.12495	0.34269	0.16369	0.41917	0.93858	0.99048	0.45507
0.13250	0.00000	0.00000	0.77944	0.08668	-0.05523	0.91402	0.93038	0.32490	-0.07058	0.99707	1.00000	0.36131	0.00000	0.89402	-0.04412	-0.05555	0.07850	0.33920	0.18472	0.45004	0.94454	0.99146	0.41780
-0.05818	0.00000	0.00000	0.12343	0.17025	-0.09003	0.18577	0.25181	-0.10416	-0.02873	0.32295	0.36131	1.00000	0.00000	0.21374	0.07086	-0.09001	-0.39475	-0.17568	-0.10778	0.10709	0.42592	0.34260	0.00387
0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
0.19456	0.00000	0.00000	0.67943	0.22988	-0.00004	0.87164	0.90173	0.55223	-0.39771	0.90139	0.89402	0.21374	0.00000	1.00000	-0.11850	-0.00041	0.20601	0.58787	0.34477	0.39786	0.80221	0.85679	0.18154
-0.01526	0.00000	0.00000	0.11142	0.01381	0.19415	0.07722	-0.05783	-0.05835	0.06111	-0.08167	-0.04412	0.07086	0.00000	-0.11850	1.00000	0.19421	-0.03142	-0.14114	-0.02186	0.02502	-0.02331	-0.02797	-0.21277
0.02067	0.00000	0.00000	-0.19594	0.02915	1.00000	-0.08622	0.07473	0.08696	-0.12737	-0.02484	-0.05555	-0.09001	0.00000	-0.00041	0.19421	1.00000	0.64450	0.10068	0.05063	-0.02930	-0.19108	-0.08074	0.50082
0.11474	0.00000	0.00000	0.16472	-0.02378	0.64486	0.03064	0.28972	0.16443	-0.18290	0.12495	0.07850	-0.39475	0.00000	0.20601	-0.03142	0.64450	1.00000	0.26677	0.06869				

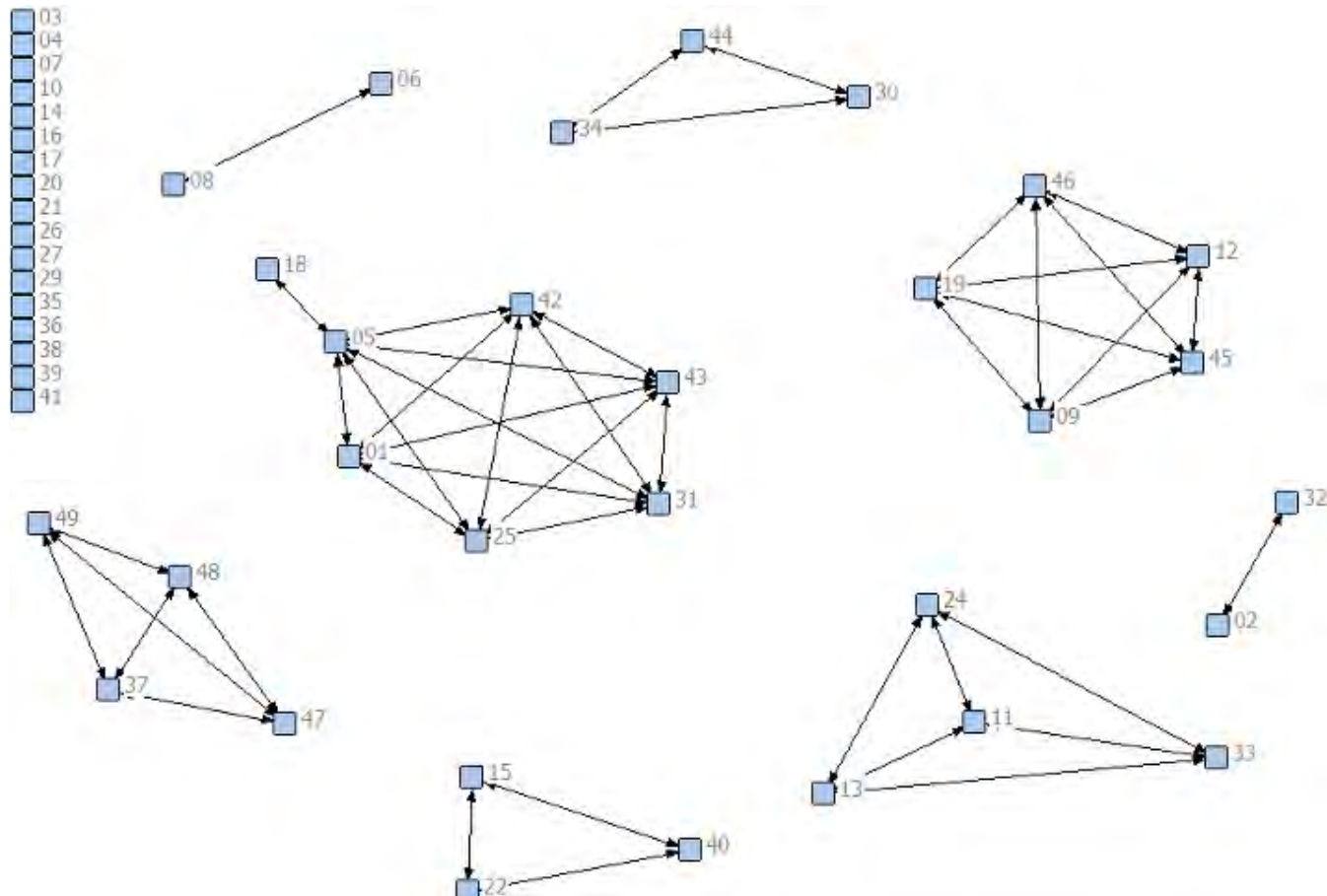
39) Rede desenhada utilizando-se os dados de similaridade gerados para o exercício 13 excluindo-se relações de similaridade menores que 0,9:



40) Matriz de similaridade para o Exercício 14 ajustado pelo fator de entropia e reduzido à $k=10$. Alunos de 1 até 24

Doc. x Doc.	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	24
01	1.00000	-0.03477	0.73078	-0.04008	0.96525	0.72871	-0.17942	0.74638	0.00321	-0.01395	0.15881	0.00321	0.15881	0.00000	0.22389	0.50167	0.71429	0.87909	0.01780	0.00000	0.08097	0.22353	0.26217
02	-0.03477	1.00000	0.04812	-0.00792	-0.04186	0.43361	-0.04406	0.41339	-0.02050	0.27581	-0.00885	-0.02050	-0.00885	0.00000	0.83588	0.27919	-0.12067	-0.07357	-0.01868	0.00000	0.05028	0.83574	0.13456
03	0.73078	0.04812	1.00000	0.04707	0.73854	0.80912	0.11521	0.80105	-0.06481	0.18247	-0.13723	-0.06481	-0.13723	0.00000	0.28623	0.31157	0.36594	0.85841	-0.04786	0.00000	-0.12694	0.28593	0.08682
04	-0.04008	-0.00792	0.04707	1.00000	-0.04076	-0.05494	-0.00259	-0.06454	-0.00193	0.00732	-0.05722	-0.00193	-0.05722	0.00000	-0.01179	-0.03067	0.08479	0.11423	-0.00445	0.00000	-0.14250	-0.01184	-0.02754
05	0.96525	-0.04186	0.73854	-0.04076	1.00000	0.57659	-0.07932	0.59677	0.03439	-0.02272	0.03356	0.03439	0.03356	0.00000	0.16231	0.46627	0.71767	0.93347	0.03698	0.00000	0.01036	0.16197	0.15076
06	0.72871	0.43361	0.50912	-0.05494	0.57659	1.00000	-0.20850	0.99800	-0.03542	0.03160	0.44162	-0.03542	0.44162	0.00000	0.59062	0.58826	0.49599	0.49697	-0.00404	0.00000	0.28688	0.59016	0.55915
07	-0.17942	-0.04406	0.11521	-0.00259	-0.07932	-0.20850	1.00000	-0.21747	0.02145	0.05166	0.42676	0.02145	0.42676	0.00000	-0.08043	-0.07179	0.29858	0.11942	0.01351	0.00000	-0.11983	-0.08063	0.49804
08	0.74638	0.41339	0.80105	-0.06481	0.59677	0.99800	-0.21747	1.00000	-0.05066	0.00154	0.44813	-0.05066	0.44813	0.00000	0.56586	0.58484	0.50888	0.50682	-0.02125	0.00000	0.28058	0.56540	0.56332
09	0.00321	-0.02050	-0.06451	-0.00193	0.03439	-0.03542	0.02145	-0.05066	1.00000	0.01013	-0.04625	1.00000	-0.04625	0.00000	-0.09206	0.73040	-0.06672	-0.03438	0.99716	0.00000	-0.00961	-0.09247	-0.01882
10	-0.01395	0.27581	0.18247	0.00732	-0.02272	0.03160	0.05166	0.00154	0.01013	1.00000	-0.11380	0.01013	-0.11380	0.00000	0.66255	-0.18389	-0.07196	-0.04955	0.05789	0.00000	-0.01436	0.66295	-0.08619
11	0.15881	-0.00885	-0.13723	-0.05722	0.03356	0.44162	0.42676	0.44813	-0.04625	-0.11380	1.00000	-0.04625	-0.11380	0.00000	0.08875	0.19528	0.58662	0.00105	-0.02431	0.00000	0.29642	0.08843	0.96503
12	0.00321	-0.02050	-0.06451	-0.00193	0.03439	-0.03542	0.02145	-0.05066	1.00000	0.01013	-0.04625	1.00000	-0.04625	0.00000	-0.09206	0.73040	-0.06672	-0.03438	0.99716	0.00000	-0.00961	-0.09247	-0.01882
13	0.15881	-0.00885	-0.13723	-0.05722	0.03356	0.44162	0.42676	0.44813	-0.04625	-0.11380	1.00000	-0.04625	-0.11380	0.00000	0.08875	0.19528	0.58662	0.00105	-0.02431	0.00000	0.29642	0.08843	0.96503
14	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
15	0.22389	0.83588	0.28623	-0.01179	0.16231	0.59062	-0.08043	0.56586	-0.09206	0.66255	0.08875	-0.09206	0.08875	0.00000	1.00000	0.20888	0.07176	0.09941	-0.06839	0.00000	0.06982	1.00000	0.21746
16	0.50167	0.27919	0.31157	-0.03067	0.46627	0.58826	-0.07179	0.58484	0.73040	0.19528	0.19528	0.73040	0.19528	0.00000	0.20888	1.00000	0.27879	0.37294	0.72812	0.00000	0.03639	0.02618	0.32298
17	0.71429	-0.12067	0.36594	0.08479	0.71767	0.49599	0.29858	0.50888	-0.06672	-0.07196	0.58662	-0.06672	0.58662	0.00000	0.07176	0.27879	1.00000	0.70063	-0.05738	0.00000	0.22085	0.07141	0.61013
18	0.87909	-0.07357	0.85841	0.11423	0.93347	0.49697	0.11942	0.50682	-0.03438	-0.04955	0.00105	-0.03438	0.00105	0.00000	0.09941	0.37294	0.70063	1.00000	-0.03177	0.00000	0.01547	0.09903	0.14961
19	0.01780	-0.01868	-0.04786	-0.00445	0.03698	-0.00404	0.01351	-0.02125	0.99716	0.05789	-0.02431	0.99716	-0.02431	0.00000	-0.05839	0.72812	-0.05738	-0.03177	1.00000	0.00000	0.03239	-0.05879	0.00153
20	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
21	0.08097	0.05028	-0.12694	-0.14250	0.01036	0.28688	-0.11983	0.28058	-0.00961	-0.01436	0.29642	-0.00961	0.29642	0.00000	0.06982	0.36339	0.22085	0.01547	0.03239	0.00000	1.00000	0.00000	0.23409
22	0.22353	0.83574	0.28593	-0.01184	0.16197	0.59016	-0.08063	0.56540	-0.09247	0.66295	0.08843	-0.09247	0.08843	0.00000	1.00000	0.20818	0.07141	0.09903	-0.05879	0.00000	0.06970	1.00000	0.21702
24	0.26217	0.13456	0.05682	-0.02754	0.15076	0.55915	0.49804	0.56332	-0.01882	-0.08619	0.96503	-0.01882	0.96503	0.00000	0.21746	0.32298	0.61013	0.14961	0.00153	0.00000	0.23409	0.21702	1.00000
25	0.98642	0.06563	0.89932	-0.01000	0.93866	0.78908	-0.18625	0.80274	-0.04195	0.01784	0.22194	-0.04195	0.22194	0.00000	0.31989	0.49329	0.75484	0.84806	-0.02518	0.00000	0.11618	0.31953	0.32088
26	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
27	0.63163	0.60372	0.55411	-0.09482	0.57423	0.86370	-0.01296	0.85918	0.24131	-0.03683	0.26104	0.24131	0.26104	0.00000	0.58194	0.80017	0.37056	0.53092	0.25080	0.00000	0.13517	0.58131	0.45675
29	-0.06899	-0.02213	-0.35538	0.06478	-0.29855	0.39900	-0.16334	0.38822	0.00778	-0.01253	0.69993	0.00778	0.69993	0.00000	0.10982	0.09483	0.12524	-0.36986	0.04999	0.00000	0.44955	0.10972	0.58370
30	-0.11252	0.74632	-0.08354	0.11634	-0.03134	0.16846	0.08586	0.16133	0.22384	-0.29777	-0.07857	0.22384	-0.07857	0.00000	0.30387	0.44667	-0.07221	-0.00809	0.18426	0.00000	-0.06346	0.30346	0.05453
31	0.99998	-0.03754	0.73077	-0.04032	0.96674	0.72452	-0.17794	0.74237	0.00437	-0.01490	0.15587	0.00437	0.15587	0.00000	0.22049	0.50084	0.71458	0.88052	0.01864	0.00000	0.07820	0.22013	0.28920
32	-0.00046	0.99880	0.07269	-0.00586	-0.01416	0.47052	-0.03834	0.44986	-0.00985	0.28628	0.02206	-0.00985	0.02206	0.00000	0.85211	0.30825	-0.08682	-0.04793	-0.00562	0.00000	0.05971	0.85194	0.16805
33	0.15881	-0.00885	-0.13723	-0.05722	0.03356	0.44162	0.42676	0.44813	-0.04625	-0.11380	1.00000	-0.04625	-0.11380	0.00000	0.08875	0.19528	0.58662	0.00105	-0.02431	0.00000	0.29642	0.08843	0.96503
34	-0.11322	0.80632	-0.09206	-0.00297	-0.08259	0.28368	0.03413	0.28167	-0.03768	-0.32159	0.02387	-0.03768	0.02387	0.00000	0.38408	0.31713	-0.10870	-0.07640	-0.07225	0.00000	-0.08048	0.38374	0.18099
35	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
36	0.18186	0.70045	0.07604	-0.03436	0.06107	0.64850	0.01885	0.62617	-0.12276	0.56469	0.47874	-0.12276	0.47874	0.00000	0.89705	0.17899	0.26023	-0.02352	-0.08013	0.00000	0.24706	0.89700	0.53564
37	0.05926	-0.02476	0.56624	-0.02031	-0.03592	0.22836	0.06702	0.20327	0.00374	-0.02585	-0.13234	0.00374	-0.13234	0.00000	0.03567	0.15180	-0.31276	0.15099	0.02058	0.00000	-0.17143	0.03545	-0.00684
38	-0.04232	-0.00789	0.00291	-0.00947	-0.02671	-0.05129	0.00194	-0.08840	0.00065	0.00363	-0.07767	0.00065	-0.07767	0.00000	-0.01392	-0.07111	0.19223	0.04531	-0.00365	0.00000	-0.00894	-0.01391	-0.14066
39	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
40	0.22353	0.83574	0.28593	-0.01184	0.16197	0.59016	-0.08063	0.56540	-0.09247	0.66295	0.08843	-0.09247	0.08843	0.00000	1.00000	0.20818	0.07141	0.09903	-0.05879	0.00000	0.06970	1.00000	0.21702
41	-0.06399	0.03076	0.08303	0.02447	-0.02036	-0.03514	0.03989	-0.04914	-0.00499	-0.00516	-0.15703	-0.00499	-0.15703	0.00000	-0.05334	-0.09708	-0.01118	0.13833	0.01966	0.00000	0.79909	-0.05341	-0.14544
42	1.00000	-0.03477	0.73078	-0.04008	0.96525	0.72871	-0.17942	0.74638	0.00321	-0.01395	0.15881	0.00321	0.15881	0.00000	0.22389	0.50167	0.71429	0.87909	0.01780	0.00000	0.08097	0.22353	0.26217
43	0.98784	-0.00143	0.66387	-0.00201	0.93869	0.76843	-0.16536	0.78495	-0.03590	-0.01679	0.26416	-0.03590	0.26416	0.00000	0.25795	0.48443	0.78523	0.84713	-0.01921	0.00000	0.13424	0.25759	0.35131
44	-0.00628	0.78009	-0.03114	-0.00572	-0.00124	0.40004	-0.00237	0.39995	-0.02937	-0.37781	0.09675	-0.02937	0.09675	0.00000	0.37968	0.39898	-0.03181	-0.00393	-0.06079	0.0000			

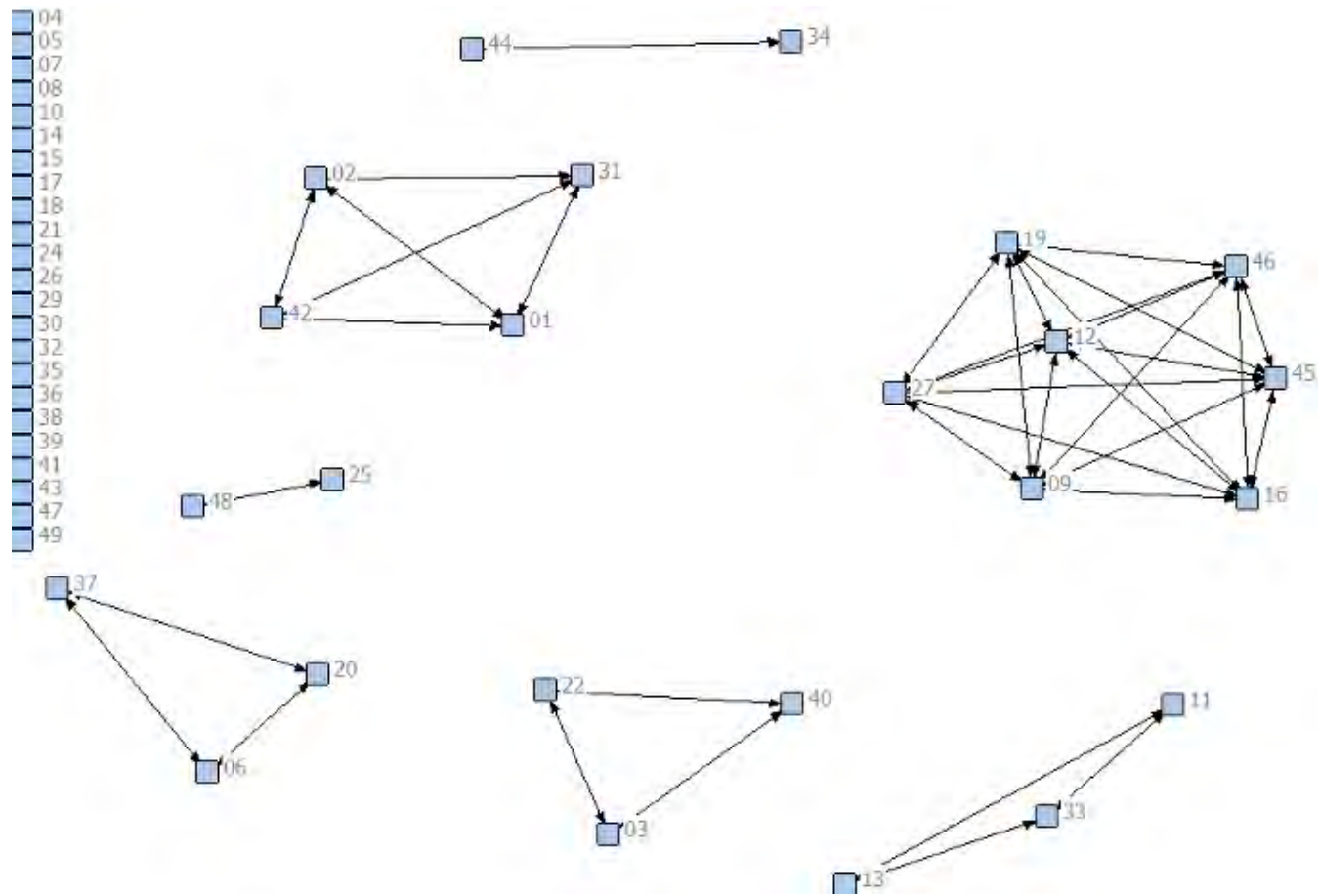
42) Rede desenhada utilizando-se os dados de similaridade gerados para o exercício 14 excluindo-se relações de similaridade menores que 0,9:



43) Matriz de similaridade para o Exercício 15 ajustado pelo fator de entropia e reduzido à $k=10$. Alunos de 1 até 24

Doc. x Doc.	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
01	1.00000	0.99824	0.02344	0.15011	0.00000	0.18181	-0.16113	0.00000	-0.06191	0.00000	0.02213	-0.06191	0.02213	0.00000	0.54413	0.10698	0.18181	-0.05636	-0.07247	0.20993	0.05197	0.02344	-0.09788	
02	0.99824	1.00000	0.05191	0.16943	0.00000	0.21187	-0.17149	0.00000	-0.04244	0.00000	0.05109	-0.04244	0.05109	0.00000	0.58150	0.13298	0.18175	-0.03608	-0.05339	0.23627	0.07348	0.05191	-0.07831	
03	0.02344	0.05191	1.00000	0.03919	0.00000	0.45505	0.26535	0.00000	0.81147	0.00000	0.31220	0.81147	0.31220	0.00000	0.67248	0.83553	0.17239	-0.09356	0.80164	0.57485	0.11750	1.00000	0.18623	
04	0.15011	0.16943	0.03919	1.00000	0.00000	0.26551	0.29158	0.00000	-0.05478	0.00000	-0.29702	-0.05478	-0.29702	0.00000	0.18316	-0.00676	-0.18619	0.46071	-0.05857	0.05480	-0.12461	0.03919	0.34272	
05	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	
06	0.18181	0.21187	0.45505	0.26581	0.00000	1.00000	0.37160	0.00000	0.12282	0.00000	0.61758	0.12282	0.61758	0.00000	0.64925	0.23960	0.01293	0.03557	0.11197	0.90272	-0.03861	0.45505	0.76419	
07	-0.16113	-0.17149	0.26535	0.29158	0.00000	0.37160	1.00000	0.00000	0.05899	0.00000	0.04492	0.05899	0.04492	0.00000	-0.15427	-0.03478	0.34096	-0.04003	0.06045	0.22047	-0.15398	0.26535	0.13473	
08	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	
09	-0.06191	-0.04244	0.81147	-0.05478	0.00000	0.12282	0.05899	0.00000	1.00000	0.00000	-0.01522	1.00000	-0.01522	0.00000	0.38181	0.97375	-0.10228	0.00764	0.99978	0.19380	-0.04040	0.81147	-0.12321	
10	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	
11	0.02213	0.05109	0.31220	-0.29702	0.00000	0.61758	0.04492	0.00000	-0.01522	0.00000	1.00000	-0.01522	1.00000	0.00000	0.48546	0.07658	0.46911	-0.13641	-0.02273	0.75251	0.06233	0.31220	0.38558	
12	-0.06191	-0.04244	0.81147	-0.05478	0.00000	0.12282	0.05899	0.00000	1.00000	0.00000	-0.01522	1.00000	-0.01522	0.00000	0.38181	0.97375	-0.10228	0.00764	0.99978	0.19380	-0.04040	0.81147	-0.12321	
13	0.02213	0.05109	0.31220	-0.29702	0.00000	0.61758	0.04492	0.00000	-0.01522	0.00000	1.00000	-0.01522	1.00000	0.00000	0.48546	0.07658	0.46911	-0.13641	-0.02273	0.75251	0.06233	0.31220	0.38558	
14	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	
15	0.54413	0.58150	0.67248	0.18316	0.00000	0.64925	-0.15427	0.00000	0.38181	0.00000	0.48546	0.38181	0.48546	0.00000	1.00000	0.56587	0.07098	-0.04759	0.36441	0.76464	0.17695	0.67248	0.38589	
16	0.10698	0.13298	0.83553	-0.00676	0.00000	0.23960	-0.03475	0.00000	0.97375	0.00000	0.07658	0.97375	0.07658	0.00000	0.56587	1.00000	-0.11043	0.01013	0.97016	0.31934	0.00191	0.83553	-0.03285	
17	0.18181	0.18175	0.17239	-0.18619	0.00000	0.01293	0.34096	0.00000	-0.10228	0.00000	0.46911	-0.10228	0.46911	0.00000	0.07098	-0.11043	1.00000	0.09317	-0.10609	0.12652	0.07609	0.17239	-0.35594	
18	-0.05636	-0.03608	-0.09356	0.46071	0.00000	0.03557	-0.04003	0.00000	0.00764	0.00000	-0.13641	0.00764	-0.13641	0.00000	0.01013	0.09317	1.00000	0.01055	-0.17466	-0.21993	-0.09356	-0.06410	0.00000	
19	-0.07247	-0.05339	0.80164	-0.06867	0.00000	0.11197	0.06045	0.00000	0.99978	0.00000	-0.02273	0.99978	-0.02273	0.00000	0.36441	0.97016	-0.10609	0.01055	1.00000	0.17970	-0.04480	0.80164	-0.13369	
20	0.20993	0.23627	0.57485	0.05480	0.00000	0.90272	0.22047	0.00000	0.19380	0.00000	0.75251	0.19380	0.75251	0.00000	0.76464	0.31934	0.12652	-0.17466	0.17970	1.00000	-0.11959	0.57485	0.74965	
21	0.05197	0.07348	0.11750	-0.12461	0.00000	-0.03861	-0.15398	0.00000	-0.04040	0.00000	0.06233	-0.04040	0.06233	0.00000	0.17695	0.00191	0.07609	-0.21993	-0.04480	-0.11959	1.00000	0.11750	-0.21408	
22	0.02344	0.05191	1.00000	0.03919	0.00000	0.45505	0.26535	0.00000	0.81147	0.00000	0.31220	0.81147	0.31220	0.00000	0.67248	0.83553	0.17239	-0.09356	0.80164	0.57485	0.11750	1.00000	0.18623	
23	-0.09788	-0.07831	0.18623	0.34272	0.00000	0.76419	0.13473	0.00000	-0.12321	0.00000	0.38558	-0.12321	0.38558	0.00000	0.43889	-0.03285	-0.35594	-0.06410	-0.13369	0.74965	-0.21408	0.18623	1.00000	
24	0.53377	0.55793	0.12913	0.65089	0.00000	0.59736	0.23945	0.00000	0.07985	0.00000	0.00091	0.07985	0.00091	0.00000	0.44660	0.20286	-0.08649	0.41302	0.07427	0.30605	0.07492	0.12913	0.26091	
25	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	
26	-0.06191	-0.04244	0.81147	-0.05478	0.00000	0.12282	0.05899	0.00000	1.00000	0.00000	-0.01522	1.00000	-0.01522	0.00000	0.38181	0.97375	-0.10228	0.00764	0.99978	0.19380	-0.04040	0.81147	-0.12321	
27	-0.06191	-0.04244	0.81147	-0.05478	0.00000	0.12282	0.05899	0.00000	1.00000	0.00000	-0.01522	1.00000	-0.01522	0.00000	0.38181	0.97375	-0.10228	0.00764	0.99978	0.19380	-0.04040	0.81147	-0.12321	
28	-0.13994	-0.10078	0.03158	-0.20452	0.00000	0.35543	-0.18549	0.00000	0.01881	0.00000	0.56870	0.01881	0.56870	0.00000	0.17192	0.07745	0.03599	0.17910	0.02244	0.21049	0.42624	0.03158	0.06154	
29	-0.07580	-0.05682	0.77232	-0.02507	0.00000	0.07386	-0.10646	0.00000	0.65782	0.00000	0.02933	0.65782	0.02933	0.00000	0.51552	0.86786	0.40449	-0.02650	0.64638	0.26430	0.19356	0.77232	0.08802	
30	0.99997	0.99787	0.02233	0.14884	0.00000	0.17739	-0.15811	0.00000	-0.06094	0.00000	0.01764	-0.06094	0.01764	0.00000	0.53946	0.10675	0.18346	-0.05694	-0.07142	0.20529	0.05024	0.02233	-0.10332	
31	0.02797	0.03902	0.18938	0.21490	0.00000	-0.05986	-0.00352	0.00000	0.00270	0.00000	-0.04198	0.00270	-0.04198	0.00000	0.16433	0.01710	0.00875	-0.01667	0.00087	0.01301	0.03936	0.18938	0.02637	
32	0.02213	0.05109	0.31220	-0.29702	0.00000	0.61758	0.04492	0.00000	-0.01522	0.00000	1.00000	-0.01522	1.00000	0.00000	0.48546	0.07658	0.46911	-0.13641	-0.02273	0.75251	0.06233	0.31220	0.38558	
33	-0.05826	-0.04305	0.53345	0.02983	0.00000	0.07643	-0.12608	0.00000	0.22930	0.00000	0.09789	0.22930	0.09789	0.00000	0.46589	0.25288	0.15261	-0.03416	0.21408	0.27239	0.29184	0.53345	0.21913	
34	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	
35	0.55977	0.58425	0.23552	-0.15119	0.00000	0.34795	-0.11805	0.00000	0.01639	0.00000	0.73817	0.01639	0.73817	0.00000	0.58386	0.15831	0.67237	0.03932	0.00758	0.47834	0.12598	0.23552	-0.06681	
36	0.20984	0.23617	0.57443	0.05480	0.00000	0.90276	0.22046	0.00000	0.19335	0.00000	0.75257	0.19335	0.75257	0.00000	0.76441	0.31889	0.12641	-0.17470	0.17925	1.00000	-0.11978	0.57443	0.74987	
37	-0.02243	-0.00323	-0.05085	0.43767	0.00000	-0.11401	-0.05671	0.00000	0.01014	0.00000	0.06440	0.01014	0.06440	0.00000	0.04212	0.03415	0.03737	-0.04858	0.01185	-0.06873	0.00447	-0.05085	-0.00317	
38	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	
39	0.02344	0.05191	1.00000	0.03919	0.00000	0.45505	0.26535	0.00000	0.81147	0.00000	0.31220	0.81147	0.31220	0.00000	0.67248	0.83553	0.17239	-0.09356	0.80164	0.57485	0.11750	1.00000	0.18623	
40	-0.07976	-0.06698	0.08038	0.26121	0.00000	-0.00244	0.22758	0.00000	-0.03326	0.00000	-0.28443	-0.03326	-0.28443	0.00000	-0.03769	-0.05084	-0.07858	-0.04595	-0.03461	-0.24252	0.82675	0.08038	-0.12734	
41	1.00000	0.99824	0.02344	0.15011	0.00000	0.18181	-0.16113	0.00000	-0.06191	0.00000	0.02213	-0.06191	0.02213	0.00000	0.54413	0.10698	0.18181	-0.05636	-0.07247	0.20993	0.05197	0.02344	-0.09788	
42	0.43894	0.42493	0.40515	-0.13255	0.00000	0.04556	0.47886	0.00000	0.26539	0.00000	0.16528	0.26539	0.16528	0.00000	0.18884	0.25325	0.77859	-0.07394	0.26093					

45) Rede desenhada utilizando-se os dados de similaridade gerados para o exercício 15 excluindo-se relações de similaridade menores que 0,9:



48) Rede desenhada utilizando-se os dados de similaridade gerados para o exercício 16 excluindo-se relações de similaridade menores que 0,9:

