

O Uso de Ambientes Gráficos para Ensino e Aprendizagem de Estruturas de Dados e de Algoritmos em Grafos

Rodrigo P. Santos^{1,2}, Heitor A. X. Costa¹, Antônio M. P. Resende¹, Jano M. Souza²

¹Grupo PqES/DCC/UFLA, Universidade Federal de Lavras, Brasil
Caixa Postal 3037 – CEP 37200-000 Lavras/MG, Brasil

²COPPE/UFRJ, Universidade Federal do Rio de Janeiro, Brasil
Caixa Postal 68511 – CEP 21945-970 Rio de Janeiro/RJ, Brasil

{rps, jano}@cos.ufrj.br, heitor@ufla.br, tonio@dcc.ufla.br

Abstract. *The process of algorithms and programming learning is considered a challenge for Computer Science students, despite the several research efforts to improve this process. The amount of abstract concepts in the first years of Computer Science courses is significant and it can be decisive for the students' motivation. This paper aims to present the results of using two tools in teaching and learning of data structures and algorithms in graphs, TBC-AED and TBC-GRAFOS. These results point for larger agility in theoretical classes (conceptual), improving the execution of algorithms visualization and propitiating space for laboratory activities.*

Resumo. *O processo de aprendizagem de algoritmos e programação é considerado um desafio para estudantes de Computação, apesar dos numerosos esforços de pesquisa para melhorar esse processo. A carga de conceitos abstratos nos primeiros anos dos cursos da área é significativa e pode ser decisiva para a motivação dos estudantes. Este artigo visa apresentar os resultados da utilização de duas ferramentas computacionais no ensino e aprendizagem de estruturas de dados e de algoritmos em grafos, o TBC-AED e o TBC-GRAFOS. Esses resultados apontam para maior agilidade em aulas teóricas (conceituais), melhorando a visualização da execução de algoritmos e propiciando espaço para atividades de laboratório.*

1. Introdução

Com o intuito de produzir melhores resultados de aprendizagem na área de Computação e facilitar a compreensão de processos abstratos, faz-se necessária a atualização de metodologias de ensino de forma geral. O contato com algoritmos e programação muitas vezes representa uma barreira significativa aos estudantes nos primeiros anos dos cursos da área [Porter e Calder, 2004]. Em parte, isso se deve à dificuldade encontrada pelos professores em acompanhar as atividades laboratoriais de programação, dado o grande número de estudantes sob sua supervisão [Tobar *et al.*, 2001]. Destaca-se, ainda, a forma de estudar (memorização de conceitos e processos) e a falta de pré-requisitos em conteúdos relacionados [Rosso e Daniele, 2000]. Essa realidade se mostra pelas altas taxas de reaprovação e de desistência de cursos, devido à diminuição da auto-estima e à geração de apatia [Haden e Mann, 2003].

Apesar de não serem recentes, esses problemas fazem parte dos grandes desafios de educação em Computação para os próximos dez anos, conforme apresentado em [McGetrick *et al.*, 2004]. Este documento destaca o fato de que os estudantes percebem os cursos na área de Computação como “dominados” pela programação e aponta o desafio da *inovação* em reconhecer e acomodar competências e habilidades de estudantes, devido à amplitude de pesquisa e de aplicação da área. Existe um desafio que contempla a questão da programação: *entender o processo de programação e a prática do programador para propiciar a transferência efetiva de conhecimentos e habilidades educacionais*, sugerindo inclusive a existência de um “quociente de programação”, análogo ao quociente de inteligência. Essas questões impactam os grandes desafios da pesquisa em Computação no Brasil, pois se concentram na busca de soluções melhor sedimentadas para o processo de formação de seus recursos humanos. Mais especificamente, artigos recentes enfatizam o problema da falta de qualidade em software e consideram sua implicação na indústria brasileira de desenvolvimento para exportação, atingindo o desafio *desenvolvimento tecnológico de qualidade* [SBC, 2006].

Diante desse contexto e de suas consequências sobre os futuros profissionais da área, destaca-se o desenvolvimento de ferramentas para apoiar o ensino de algoritmos e programação [Pimentel *et al.*, 2003]. A fim de facilitar a visualização das abstrações de estruturas de dados e de grafos e propiciar atividades práticas de laboratório, o objetivo deste artigo é apresentar os resultados do uso de duas ferramentas computacionais no ensino e aprendizagem de estruturas de dados e de algoritmos em grafos, o TBC-AED e o TBC-GRAFOS. Além disso, almeja-se reduzir o índice de reprovação das disciplinas que englobam os tópicos abordados pelas ferramentas.

O artigo está organizado da seguinte forma: a seção 2 discorre sobre a fundamentação teórica; a seção 3 lista alguns trabalhos relacionados; a seção 4 faz uma breve apresentação das ferramentas TBC-AED e TBC-GRAFOS; a seção 5 descreve o uso das ferramentas e mostra os resultados obtidos; e a seção 6 apresenta as conclusões.

2. Fundamentação Teórica

Nos primeiros semestres de cursos de Computação, ocorre o contato com as disciplinas de algoritmos e programação, que preparam o estudante para atuar em áreas específicas, e isso pode gerar afinidade ou repulsa [Baeza-Yates, 2000]. Buzin (2001) mostra que os problemas de compreensão de conceitos abstratos se originam durante a formação básica: os novos discentes cursaram disciplinas ao longo da vida baseadas no paradigma de apresentação de problemas e soluções, ao invés de focar no processo de questionamento. Uma vez que esses estudantes precisam adquirir a capacidade de aprendizagem independente, as disciplinas de Computação devem incitar o desenvolvimento do raciocínio crítico e das habilidades cognitivas para a solução de problemas [Ben-Ari, 2001].

Outros problemas relacionados à aprendizagem de algoritmos e programação pelos discentes são: (i) a falta de motivação do estudante, devido ao despreparo e ao desânimo diante da crença que a disciplina constitui um obstáculo difícil de ser superado [Haden e Mann, 2003]; (ii) o processo tradicional de avaliação, que segue o modelo dos cursos de natureza teórica-prática e presencial (provas periódicas e trabalhos de programação) e que pode deixar o estudante tenso e prejudicar seu aprendizado [Santos *et al.*, 2008a]; (iii) o relacionamento entre professor e estudante, quando o primeiro se preocupa em mostrar o que sabe, desconsiderando um ambiente de aprendizagem des-

contraído e colaborativo [Winslow, 1996]; (iv) a didática ou a falta de metodologia de ensino [Santos e Costa, 2005]; e (v) a falta de comunicação entre os professores das várias disciplinas almejando tornar o trabalho mais integrado [Pimentel *et al.*, 2003].

Por outro lado, Menezes e Nobre (2002) apontam dificuldades vivenciadas pelo professor: (i) reconhecer as habilidades inatas dos estudantes; (ii) apresentar técnicas para solução de problemas; (iii) trabalhar a capacidade de abstração do estudante, na busca das possíveis soluções e na escolha da estrutura de dados a ser utilizada; e (iv) promover a cooperação e a colaboração [Hübscher-Younger e Narayanan, 2003]. Vizando melhorar essa realidade, Garcia *et al.* (1997) afirmam que diversas ferramentas de animações de algoritmos foram produzidas, explorando o potencial da visualização gráfica das operações realizadas e baseando-se no fato de que movimentos em tempo real, cores e sons enriquecem o poder de comunicação e aprendizagem.

Por fim, Setubal (2000) apresenta algumas recomendações: (i) *coerência com os objetivos fundamentais*: professores devem expressar claramente suas idéias, conceitos e técnicas diante dos estudantes, ressaltando os resultados teóricos e o rigor formal na solução de problemas; (ii) *ênfase no pensamento crítico*: estudantes devem ser estimulados a questionar demonstrações matemáticas; e (iii) *a teoria na prática*: o uso de exemplos do mundo real sempre que possível, como recurso didático, e projetos de implementação, a fim de tornar as disciplinas menos abstratas, são importantes.

3. Trabalhos Relacionados

Há diversas ferramentas construídas para o ensino de algoritmos e programação. No entanto, cada trabalho apresenta um foco relacionado às necessidades de seus desenvolvedores e professores interessados. Analisando esses trabalhos e adicionando as experiências adquiridas em disciplinas de estruturas de dados e de algoritmos em grafos, definiu-se um conjunto de requisitos para que ambientes gráficos de execução passo a passo pudessem servir como ferramentas para auxílio ao ensino e aprendizagem. A seguir, são apresentados alguns desses trabalhos. Uma avaliação de usabilidade de alguns deles (ressaltando os diferenciais e as vantagens do TBC-AED e do TBC-GRAFOS) pode ser obtida em [Santos *et al.*, 2006].

ASTRAL (Animation of Data Structures and Algorithms) [Garcia *et al.*, 1997] consiste em um ambiente para implementação de estruturas de dados e de seus algoritmos, no qual é possível visualizá-las graficamente por meio de animações que mostram a atuação dos algoritmos sobre elas. No entanto, apresenta alguns problemas: (i) ausência de informações sobre os passos dos algoritmos em execução e os elementos visualizados; (ii) em algoritmos de ordenação, uso de pontos dispersos durante a animação, ao invés de números; (iii) execução rápida e contínua, que pode causar estresse ao usuário, e a falta de metáforas para representar botões e recursos de interação; e (iv) os problemas ocorridos durante a interação com o usuário são listados, mas sem exibir qualquer solução ou idéia. *EDDL* (Estruturas de Dados Dinâmicas Lineares) [Azul e Mendes, 1998] é um ambiente que engloba: (i) uma abordagem introdutória aos conceitos ou especificação dos tipos de estruturas de dados; (ii) as técnicas de implementação; e (iii) o uso dessas estruturas na resolução de problemas. Oferece um conjunto de ferramentas de programação que o revelam ser intuitivo e de fácil manipulação, eficaz para desenvolver projetos quando o aspecto gráfico é importante. No entanto, EDDL apresenta somente versão para Windows e contempla apenas os tópicos relacionados a listas linea-

res, pilhas e filas, além de apresentar informações e textos em demasia sem utilizar a divisão da tela em *frames* (desviando a atenção do usuário).

DIDAGRAPH [Dagdilelis e Stratzemi, 1998] permite o acompanhamento de animações da execução de algoritmos, em alto nível, por meio de uma visualização do grafo com o algoritmo em andamento e de uma descrição em linguagem de alto nível do algoritmo. Possui alguns problemas: (i) ausência de textos explicativos sobre o *status* de execução; (ii) ausência de metáforas para representar botões e recursos de interação; e (iii) execução contínua (não permite paralisar o processo). *EVEGA* [Khuri e Holzapfel, 2001] consiste em uma ferramenta para o aprendizado de algoritmos em grafos e fornece uma interface poderosa e de boa usabilidade para criação e edição interativa de grafos e animação de algoritmos. Permite a comparação das execuções de vários algoritmos em gráficos de desempenho. Um algoritmo de fluxo máximo acompanha a ferramenta para demonstração, mas é possível incluir novos algoritmos. A operação do EVEGA, para geração das animações, é feita de forma intrusiva no código do algoritmo, podendo ser necessário dividir a atenção do problema a ser solucionado com a criação da animação.

4. TBC-AED e TBC-GRAFOS

Visando incorporar maior agilidade em aulas teóricas ao melhorar a visualização da execução de algoritmos e propiciar espaço para as atividades laboratoriais, foram desenvolvidas as ferramentas educacionais *TBC-AED* (Treinamento Baseado em Computador para Algoritmos e Estruturas de Dados) e *TBC-GRAFOS* (Treinamento Baseado em Computador para Algoritmos em Grafos). Essas ferramentas consistem em produtos de software educacionais relativos a tópicos básicos de algoritmos e programação, com conteúdo teórico sintético, acompanhado de execução gráfica passo a passo, parte de um repositório didático de ambientes de visualização gráfica de algoritmos abordados em disciplinas da área de Computação. Suas principais metas são: (i) facilitar a visualização e o entendimento das informações apresentadas; (ii) disponibilizar o tempo gasto com quadro-negro para a resolução de exercícios de aplicação e fixação; e (iii) alcançar maior interação entre professor e estudantes, ampliando o espaço para questionamentos.

TBC-AED e *TBC-GRAFOS* possuem características que facilitam seu uso, conforme verificado em [Santos *et al.*, 2006]: (i) utilização de linguagem simples; (ii) exibição dos botões necessários a cada momento de utilização das ferramentas; (iii) *links* explicativos, que evitam a necessidade de tutorial; (iv) processo gráfico passo a passo com elementos numéricos, melhorando a visualização e o entendimento; (v) legendas explicativas, que ilustram as etapas de execução dos algoritmos; e (vi) conteúdo teórico simples, visando familiarizar o estudante com o assunto.

O *TBC-AED* (Figura 1) aborda os seguintes tópicos: (i) Busca Binária e Métodos de Ordenação (*Select Sort*, *Insert Sort*, *Bubble Sort*, *Merge Sort* e *Quick Sort*); (ii) Estruturas de Alocação Estática e Dinâmica de Memória (Lista, Fila e Pilha); e (iii) Árvore Binária de Busca. O *TBC-GRAFOS* (Figura 2) apresenta: (i) buscas em grafos (percurso em profundidade e em largura); (ii) árvore geradora mínima (algoritmos de Kruskal e de Prim); e (iii) caminho mínimo entre vértices (algoritmos de Dijkstra e de Bellman-Ford). Ambas as ferramentas tiveram sua versão adaptada para *web* (*applets Java*), gerando *TBC-AED/WEB* e *TBC-GRAFOS/WEB*, respectivamente, a fim de alcançar acessibilidade e interatividade com usuários diversos e evitar a necessidade de *download* das ferramentas. Mais detalhes em relação à funcionalidade e à interface grá-

fica do TBC-AED e do TBC-GRAFOS podem ser obtidos em [Santos e Costa, 2005] e [Santos e Costa, 2006], respectivamente.

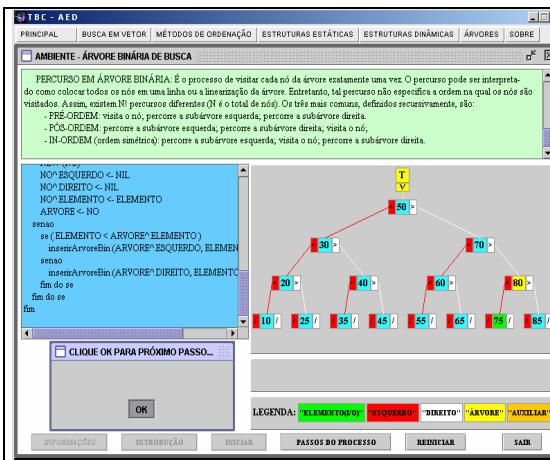
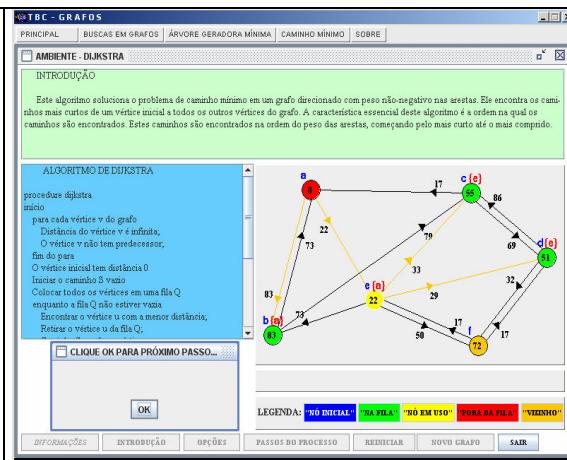


Figura 1 – Tópico Árvore Binária de Busca (TBC-AED)



em utilizá-las continuamente. Devido aos resultados positivos e visando melhorá-los, foi sugerido dividir a carga horária das disciplinas (antes apenas teórica) em aulas expositivas (teóricas) e de laboratório (uso da ferramenta e exercícios de programação) no 2º semestre de 2006. O quadro docente das disciplinas não teve alteração e os professores envolvidos se reuniram e elaboraramementas e cronogramas das disciplinas, juntamente com os autores das ferramentas. Procurou-se manter o mesmo nível das avaliações e dos trabalhos, sobretudo pela continuidade dos professores das disciplinas. Ocorreram novamente dois seminários de apresentação das ferramentas, para aplicação de questionários antes e após o seu uso. Os autores das ferramentas acompanharam as aulas para coletar observações e resolver quaisquer problemas. Além disso, os algoritmos e os tópicos teóricos relacionados foram explicados nas ferramentas, sem prévias em aulas teóricas, e os estudantes eram lembrados da importância desse processo e incentivados a estudar nas ferramentas, pois elas continham textos relativos a conceitos abordados.

Após o uso das ferramentas durante o 2º semestre de 2006, aplicou-se um questionário e os dados coletados foram tabulados para verificar a sua aplicabilidade e os resultados obtidos, entre 10/04/2007 e 10/07/2007. O questionário continha 9 questões e abordava desde aspectos de interface gráfica até a avaliação da disciplina com o uso das ferramentas, incluindo uma questão sobre o uso extra-classe das ferramentas para estudo e sugestões acerca de fatores relevantes para a melhoria das aulas. Utilizou-se a escala nominal para as respostas: ótimo/bom/regular (questões 1-7) ou sim/talvez/não (questão 8). A turma da disciplina AED II possuía 48 estudantes. Destes, 37 preencheram o questionário, sendo 15 estudantes repetentes e 22 não. Além disso, esses estudantes pertenciam do 2º ao 7º período do curso de Ciência da Computação. A turma de TP apresentava 17 estudantes, sendo que 13 preencheram o questionário e não eram estudantes repetentes; esses estudantes pertenciam do 3º ao 5º período. A seguir, são apresentadas as questões respondidas pelos estudantes e os principais resultados obtidos após a análise dos dados. No questionário, havia campos textuais em cada uma das perguntas, para observações, além da própria questão 9 (relativa a sugestões). Notou-se que aproximadamente 80% dos estudantes preencheram esses campos, que representa um fator importante de motivação e de *feedback*. A Figura 3 e a Figura 4 apresentam os resultados das questões 1-7 e a Figura 5 corresponde ao resultado da questão 8.

Questão 1 – aspectos de interface gráfica: os resultados obtidos apresentam respostas do tipo “ótimo” ou “bom” para 90% (TBC-GRAFOS) e 100% (TBC-AED) dos casos, demonstrando que a interface gráfica parece ser intuitiva e visualmente agradável. Ocorreram algumas observações referentes à inclusão de um botão para retroceder passos de execução de algoritmos, correção de problemas na barra de rolagem durante a visualização dos pseudocódigos e salvamento das instâncias de dados utilizadas pelos usuários.

Questão 2 – compreensão das telas e facilidade de utilização: as respostas do tipo “ótimo” superaram 90% para o TBC-AED e 80% para o TBC-GRAFOS. Isso reflete a preocupação anterior dos autores das ferramentas em mantê-las mais transparente possível com relação ao seu funcionamento, buscando concentrar a atenção dos usuários na aprendizagem e evitando necessidade de treinamento e/ou tutorial para sua utilização.

Questão 3 – conteúdos abordados pela ferramenta: os resultados foram do tipo “ótimo” ou “bom”, quase em igual proporção, para ambas as ferramentas. O índice considerável da resposta “bom” se deve ao fato dos estudantes sugerirem a integração de mais tópicos

das disciplinas nas ferramentas, como balanceamento em árvores binárias, tabelas de dispersão, árvores B e B* e algoritmos de fluxo máximo em grafos. Isso indica que a inclusão de *links* para animações de algoritmos disponíveis na web como recurso complementar é interessante, enquanto as ferramentas não englobam esses tópicos.

Questão 4 – qualidade dos textos apresentados: respostas do tipo “bom” aconteceram em mais de 50% dos casos. Pelo fato de terem sido concebidas para apoiar o ensino e aprendizagem de algoritmos e programação, as ferramentas possuem textos sintéticos, considerados essenciais para que o estudante não se prenda muito a elas, mas pesquise e utilize livros e outros materiais para realizar um estudo mais amplo. Ao passo que um estudante afirmou que “*deveria se cuidar para que o aluno não se acomode, dada a apresentação de conteúdos mastigados*”, ocorreram diversos pedidos para ampliação do conteúdo teórico e inserção de exercícios nas ferramentas.

Questão 5 – avaliação da disciplina sem utilização da ferramenta (para tópicos não abordados pela ferramenta): considerando os esforços dos docentes em facilitar a apresentação dos algoritmos, mais de 80% dos estudantes responderam “bom” para o ensino dos algoritmos e conteúdos não abordados pelas ferramentas. Esse fato é consequência da questão 3, na qual os estudantes solicitavam maior cobertura de conteúdos pelas ferramentas. Nessa ocasião, algumas observações também se referiam à inclusão de testes de mesa contemplando as variáveis dos algoritmos tratados pelas ferramentas.

Questão 6 – avaliação da disciplina com utilização da ferramenta (para tópicos abordados pela ferramenta): os estudantes da disciplina AED II responderam “bom” para o uso do TBC-AED durante as aulas, em mais de 60% casos. Por outro lado, aqueles que cursaram a disciplina TP responderam “ótimo” para o uso do TBC-GRAFOS, em 70% dos casos. Pode-se interpretar esse fato como consequência do número de estudantes (48 na primeira disciplina e 17 na segunda), onde o acompanhamento dos aprendizes se fez melhor na turma menor, a qual não possuía estudantes repetentes. Na turma maior, na qual havia quase 50% de repetentes, talvez o fato de rever conteúdos pode ter causado desestímulo. Além disso, soma-se o fator de primeira experiência de aplicação contínua das ferramentas. No entanto, esses resultados eram esperados.

Questão 7 – avaliação da idéia de inserção de ferramentas computacionais para ensino e aprendizagem de algoritmos e programação: após as experiências de uso das ferramentas, os estudantes responderam “ótimo” para iniciativas desse tipo, em 80% dos casos na turma que utilizou o TBC-AED e em 100% dos casos na turma que utilizou o TBC-GRAFOS. Este último caso mostra que, com a redução do índice de reprovação e consequente geração de turmas menores, pode-se esperar melhores resultados com a inserção das ferramentas, motivando os professores a continuarem com seu uso.

Questão 8 – utilização da ferramenta para estudo extra-classe: considerou-se as opiniões dos estudantes fundamentais nesse caso. 68% dos estudantes da disciplina AED II e 92% dos estudantes da disciplina TP afirmaram que utilizaram as ferramentas TBC-AED e TBC-GRAFOS para estudo extra-classe, respectivamente. 27% dos estudantes de AED II e 8% de TP afirmaram que utilizaram as ferramentas poucas vezes (“talvez”). Ainda, para a disciplina AED II, 5% dos estudantes disseram que não utilizaram o TBC-AED. Neste caso, as principais justificativas mencionadas foram tempo e falta de interesse. Pela análise dos dados, percebeu-se que esses estudantes haviam cursado a disciplina anteriormente, o que pode ter interferido como um fator desestimulante.

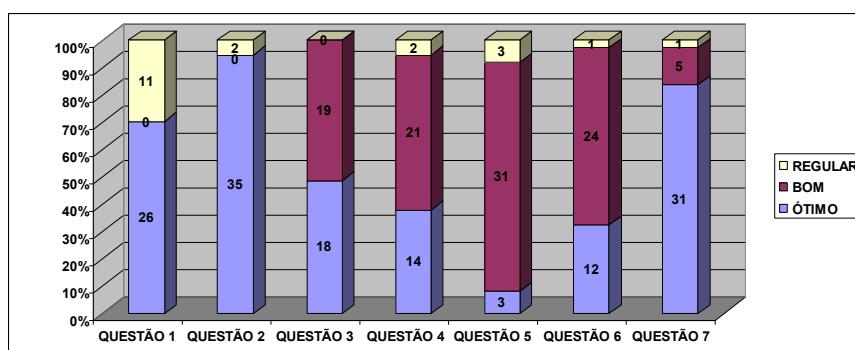


Figura 3 – Resultados extraídos do questionário sobre o uso do TBC-AED

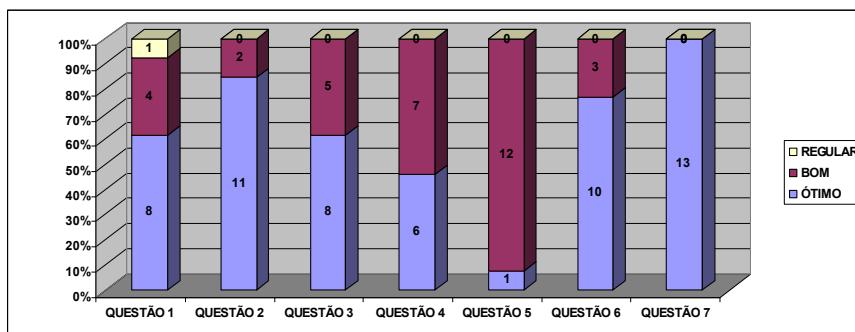


Figura 4 – Resultados extraídos do questionário sobre o uso do TBC-GRAFOS

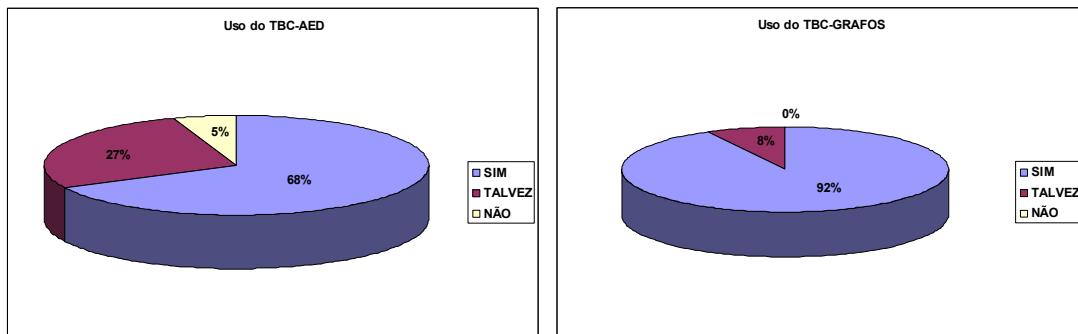


Figura 5 – Resultados sobre o uso das ferramentas para estudo extra-classe

Questão 9 – sugestões de fatores relevantes para a melhoria das aulas: (i) uso de uma linguagem de programação ao invés de pseudocódigo; (ii) correção de bugs; (iii) possibilidade de inserção de valores para os algoritmos de ordenação (e não serem aleatórios); e (iv) detalhamento das variáveis dos algoritmos. Um ponto interessante consistiu no interesse de alguns estudantes em participar do projeto de construção de ferramentas para ensino de algoritmos para outras disciplinas de Computação. Isso reforça os resultados da questão 7, destacando o aspecto motivacional, fundamental para o desenvolvimento das habilidades e conhecimentos dos estudantes.

Considerando a disciplina TP, dos 17 estudantes, 14 foram aprovados, 2 desistiram da disciplina (estudantes que estavam há mais de 4 anos no curso) e apenas um foi reprovado, gerando um índice de reprovação e desistência de aproximadamente 17%, melhor em relação à situação existente. No entanto, a média das notas finais dos estudantes foi 64, mostrando a necessidade de atuar na melhoria de desempenho para as próximas turmas. Por outro lado, com relação à disciplina AED II, com 48 estudantes,

27 foram aprovados e 21 foram reprovados, levando a um índice de reprovação de 44%. A média das notas finais foi 48. Apesar do caráter negativo desses resultados, o índice de 44% se mostra relativamente melhor do que os índices obtidos antes do uso do TBC-AED (até 70%). Uma estratégia para reduzir esse índice e para melhorar o desempenho dos estudantes deveria contemplar a redução do número de estudantes por turma, gerando turmas de, no máximo, 25 estudantes. Além disso, verificou-se que a incorporação das ferramentas melhorou a visualização gráfica dos algoritmos e possibilitou a obtenção de espaço para atividades de laboratório, incluindo exercícios práticos para fixação de conteúdo.

6. Conclusão

Diante da problemática envolvendo o ensino de algoritmos e programação, percebe-se que sua importância foi e é reconhecida pelas pesquisas realizadas, pois esses conteúdos necessitam de métodos de aprendizagem diferentes dos tradicionalmente empregados [McGetrick *et al.*, 2004]. Essas pesquisas tornam-se decisivas frente aos grandes desafios da pesquisa em Computação no Brasil, sobretudo provendo uma melhor formação de recursos humanos para o *desenvolvimento tecnológico de qualidade*.

Pelo fato do uso de ferramentas para ensino de algoritmos e programação consistir em uma idéia inteligente, se for amadurecida e desenvolvida de forma cuidadosa e estruturada, este trabalho contribui com a apresentação dos resultados obtidos com o uso do TBC-AED e do TBC-GRAFOS. Os resultados demonstram que as ferramentas facilitaram a visualização das abstrações dos algoritmos implementados e que seu uso propiciou atividades práticas de laboratório. Além disso, notou-se uma redução no índice de reprovação da disciplina TP, que não continha estudantes repetentes e possuía menos de 20 estudantes. Por outro lado, a disciplina AED II, com quase 50 estudantes (aproximadamente 50% repetentes), apresentou um índice de reprovação relativamente alto (44%), mas considerado melhor em relação ao histórico de semestres anteriores. Isso pode ter ocorrido devido ao elevado número de estudantes e ao fato de primeira experiência de utilização contínua do TBC-AED pelos professores.

A partir desses resultados, decidiu-se manter o uso do TBC-AED e do TBC-GRAFOS e montar turmas de no máximo 25 estudantes. Novos dados estão sendo coletados e serão tabulados visando extrair novos resultados acerca de novos fatores ajustados. Além disso, visando obter um tratamento personalizado dos estudantes e incorporar maior colaboração, cooperação e coordenação ao processo de ensino e aprendizagem de algoritmos e programação, novas pesquisas estão sendo realizadas para o desenvolvimento de um ambiente que apóie esse processo [Santos *et al.*, 2008b].

Agradecimentos

Os autores agradecem a FAPEMIG e ao CNPq pelo apoio financeiro para realizar este trabalho.

Referências

- Azul, A. A.; Mendes, A. J. (1998) “EDDL: Um Programa Didático sobre Estruturas de Dados Dinâmicas Lineares”, 3º Simpósio Investigação e Desenvolvimento de Software Educativo, Évora, Portugal.
- Baeza-Yates, R. A. (2000) “Teaching Algorithms”, In: SIGACT News 26, v. 4, p. 51-59.
- Ben-Ari, M. (2001) “Constructivism in Computer Science Education”, In: Journal of Computers in Mathematics and Science Teaching, v. 20, n. 1, p. 45-73.

- Buzin, P. F. W. K. (2001) “A epistemologia da Ciência da Computação: Desafio do Ensino dessa Ciência”, In: Revista de Educação, Ciência e Cultura – Centro Universitário La Salle, v. 6, n. 2, Canoas/RS.
- Dagdilelis, V.; Stratzem, M. (1998) “DIDAGRAPH: Software for Teaching Graph Theory Algoritms”, In: Proc. of the 3rd ITiCSE, Dublin, Ireland, p. 64-68.
- Garcia, I. C.; Rezende, P. J.; Calheiros, F. C. (1997) “Astral: Um Ambiente para Ensino de Estruturas de Dados através de Animações de Algoritmos”, In: RBIE, n. 1.
- Haden, P.; Mann, S. (2003) “The Trouble with Teaching Programming”, In: Proc. of the 16th Annual NACCQ, Palmerston North, New Zealand, p. 63-70.
- Hübscher-Younger, T.; Narayanan, N. H. (2003) “Constructive and Collaborative Learning of Algorithms”, In: ACM SIGCSE Bulletin, v. 35, n. 1, p. 6-10.
- Khuri, S.; Holzapfel, K. (2001) “EVEGA: An Education Visualization Environment for Graph Algorithms”, In: Proc. of the 6th ITiCSE, Canterbury, UK, p. 101-104.
- McGetrick, A.; Boyle, R.; Ibbett, R.; Lloyd, J.; Lovegrove, G.; Mander, K. (2004) “Grand Challenges in Computing – Education”. The British Computer Society, 26p.
- Menezes, C.; Nobre, I. (2002) “Suporte à Cooperação em um Ambiente de Aprendizagem para Programação (SambA)”, In: XIII SBIE, São Leopoldo/RS, p. 337-347.
- Pimentel, E. P.; França, V. F.; Omar, N. (2003) “Avaliação Contínua da Aprendizagem, das Competências e Habilidades em Programação de Computadores”, In: Anais do IX WEI, CSBC’2003, Campinas/SP, p. 105-116.
- Porter, R.; Calder, P. (2004) “Patterns in Learning to Program – An Experiment?”, In: Australasian Computing Education Conference, Dunedin, New Zealand, p. 241-246.
- Rosso, A.; Daniele, M. (2000) “Our Method to Teach Algorithmic Development”, In: ACM SIGCSE Bulletin, v. 32, n. 2, ACM Press, USA, p. 49-52.
- Santos, R. P.; Costa, H. A. X. (2005) “TBC-AED e TBC-AED/WEB: Um Desafio no Ensino de Algoritmos, Estruturas de Dados e Programação”, In: Anais do IV WEIMIG, Varginha/MG.
- Santos, R. P.; Costa, H. A. X. (2006) “Um Software Gráfico Educacional para o Ensino de Algoritmos em Grafos”, In: Proc. of the IADIS/CIAWI’2006, Murcia, Spain, p. 358-362.
- Santos, R. P.; Costa, H. A. X.; Zambalde, A. L. (2006) “Avaliação de Interfaces de Ferramentas Computacionais para o Ensino de Estruturas de Dados e Algoritmos em Grafos: Heurísticas de Usabilidade”, In: Anais do V WEIMIG, Ouro Preto/MG.
- Santos, R. P.; Vivacqua, A. S.; Souza, J. M.; Costa, H. A. X. (2008a) “Questões e Desafios no Ensino de Algoritmos e Programação: Identificando Aspectos Importantes no Processo de Aprendizagem apoiado por Computador”, In: Proc. of the X INTERTECH, Peruíbe/SP, p. 568-572.
- Santos, R. P.; Vivacqua, A. S.; Souza, J. M.; Costa, H. A. X. (2008b) “Uma Proposta de Cenário para Ensino de Algoritmos e Programação com Contribuições de Cooperação, Colaboração e Coordenação”, In: Anais do XVI WEI, CSBC’2008, Belém/PA.
- SBC (2006) “Grandes Desafios da Pesquisa em Computação no Brasil – 2006-2016”.
- Setubal, J. C. (2000) “Uma proposta de Plano Pedagógico para a Matéria de Computação e Algoritmos”, In: II Curso de Qualidade de Cursos de Graduação da Área de Computação e Informática (WEI’2000). Editora Universitária Champagnat.
- Tobar, C. M.; Rosa, J. L. G.; Coelho, J. M. A.; Pannain, R. (2001) “Uma Arquitetura de Ambiente Colaborativo para o Aprendizado de Programação”, In: Anais do XII SBIE, Vitória/ES, p. 21-23.
- Winslow, L. E. (1996) “Programming Pedagogy – A Psychological Overview”, In: ACM SIGCSE Bulletin, v. 28, n. 3, p. 17-22.