

Análise de Metodologias e Ambientes de Ensino para Algoritmos, Estruturas de Dados e Programação aos iniciantes em Computação e Informática*

Rodrigo Pereira dos Santos¹
Heitor Augustus Xavier Costa²

UFLA – Universidade Federal de Lavras
DCC – Departamento de Ciência da Computação
Cx Postal 37 – CEP 37200-000 – Lavras (MG)
¹rpsantos@comp.ufla.br
²heitor@ufla.br

Resumo. Este artigo tem o objetivo de apresentar a importância de um bom ensino de fundamentos de algoritmos e programação em cursos de graduação em Computação e Informática. Para isso, serão listados alguns problemas de ensino de tais conteúdos, juntamente com algumas estratégias pesquisadas na literatura para tentar saná-los. Dessa forma, tenta deixar uma contribuição à literatura e aos docentes da área.

Palavras-chaves: Metodologias de Ensino de Computação, Algoritmos e Estruturas de Dados, Programação.

Methodologies Analysis and Environment of Teaching for Algorithms, Data Structures and Programming to beginners in Computing

Abstract. This paper has the objective to present the importance of a good teaching of algorithms and programming foundations in graduation courses in Computing. For that, some problems of teaching of such contents will be presented, together with some strategies researched in the literature to try solving them. In that way, it tries to leave a contribution to the literature and the teachers of the field.

Key-words: Computing Teaching Methodologies, Algorithms and Data Structures, Programming.

(Received December 07, 2005 / Accepted January 24, 2006)

1. Introdução

A crescente quantidade de informação e a tecnologia gerada atualmente tornam necessária a presença de profissionais de caráter persuasivo e inovador, que sejam capazes de unir inteligência e dinamismo na busca por melhores soluções de problemas. Isso encontra sua principal base nas metodologias de ensino aplicadas durante o estágio universitário, principalmente na área de Computação e Informática, que envolve tecnologias em constante desenvolvimento.

Com o intuito de produzir melhores resultados no processo de aprendizagem nessas áreas, faz-se constante a necessidade de atualização das didáticas de ensino de forma geral. O objetivo disso é transformar processos

abstratos em concretos, através de produtos de *software* que possibilitem melhor compreensão dos tópicos abordados e maior interação entre aluno-objeto de trabalho (neste caso, o computador). Isso encontra sua maior barreira nos estágios iniciais dos cursos relacionados, quando do contato com o estudo de **algoritmos, raciocínio lógico e programação**.

Segundo [6], algoritmo é a descrição de uma seqüência de passos que deve ser seguida para a realização de uma tarefa e ação é um acontecimento que, a partir de um estado inicial, após um período de tempo finito, produz um estado final previsível e bem definido. Assim, um algoritmo é a descrição de um conjunto de comandos que, obedecidos, resultam em uma sucessão finita de ações.

* Apoio PBIICT/FAPEMIG

Atualmente, o ensino de algoritmos busca nas Ciências Exatas seu pilar de sustentação, pois disciplinas nessa área despertam o raciocínio matemático-lógico para resolução de problemas. A título de exemplo, conforme [14], alguns autores preconizam a necessidade de domínio de habilidades matemáticas prévias ou ao menos integradas, ou pelo menos desejável, sendo considerado até um bom indicador de sucesso para o processo. Para outros, o próprio nível de conhecimento prévio de lógica matemática é discutido. Mas, isso deve ser mais uma das ferramentas para o estudante e não a sua única fonte, pois a sua formação futura será muito ampla.

O conceito de programação, segundo [26], é a arte ou a técnica de construir e formular algoritmos de uma forma sistemática. Esse conceito induz que o ingresso tem um dom (arte) de programar ou ele pode aprender as técnicas necessárias a aprimorar seu conhecimento. Logo, ao ingressar em um curso superior na área, ele deve ter sua base bem sedimentada, para que fique apto a prosseguir de maneira positiva durante os seus estudos. Com isso, professores de disciplinas relacionadas à programação e coordenadores de curso sentem a grande responsabilidade de buscar e aperfeiçoar sua maneira de conduzir a estrutura disciplinar e curricular dos graduandos em seus estágios iniciais da universidade.

Com toda esta bagagem a ser assimilada, uma linha de pesquisa, visando melhorias no ensino básico em Computação e Informática, é aquela direcionada para a tendência de utilizar ferramentas computacionais como ambientes de estudo. Segundo [24], a partir de observações dentro de disciplinas de graduação, percebe-se também um melhor resultado no aprendizado por meio de atividades práticas de desenvolvimento de simuladores e ferramentas visuais didáticas de representação de conceitos abstratos. Além disso, acompanhando esse desenvolvimento, a linha acopla contribuições dos pesquisadores para novas estratégias de transmissão de conhecimentos. Isso promove a ampliação do acervo existente, a melhor organização de currículos para o mercado de trabalho e o acesso facilitado de tais recursos inovadores através de suporte via *web*.

No entanto, a elaboração de uma ferramenta computacional didática deve tornar o ensino do conteúdo abordado mais prático e abrangente, de forma a despertar o interesse do aluno, o seu espírito de pesquisa e a busca de informações que possam torná-lo um profissional crítico e de opinião sólida. Assim, seu rendimento ao longo do curso aumentaria, melhorando o seu desempenho em disciplinas mais específicas. Além disso, deve despertar a atenção dos alunos, pela inovação e dinamização injetadas às aulas, com sua utilização ([20]).

Este artigo tem por objetivo apresentar a importância de um bom ensino de fundamentos de algoritmos e programação em cursos de graduação em

Computação e Informática, listando alguns problemas de ensino de tais conteúdos, juntamente com algumas estratégias pesquisadas na literatura para tentar saná-los. Isso possibilita maior disseminação do conhecimento e promove a interatividade entre docentes e graduandos da área, enriquecendo a pesquisa com críticas e sugestões. Para alcançar este objetivo, são relatados alguns ambientes existentes de ensino a distância.

O artigo está organizado da seguinte forma: a seção 2 apresenta a importância da programação e dos algoritmos no ensino de Computação; a seção 3 apresenta um breve relato sobre os problemas acerca do ensino de Computação, listando algumas pesquisas, com soluções e considerações; a seção 4 retrata a importância do uso de ferramentas computacionais no ensino, com suas vantagens; e a seção 5 detalha algumas ferramentas disponíveis atualmente para o ensino de algoritmos, estruturas de dados e programação. Por fim, na seção 6, são apresentadas conclusões.

2. Programação e Algoritmos: a Base da Computação

Segundo a estrutura curricular constante nas Diretrizes Curriculares do MEC ([1]), a matéria de Programação faz parte da área de formação básica em Ciência da Computação, juntamente com as matérias Computação e Algoritmos e Arquitetura de Computadores. Seu conteúdo abrange, além do ensino de linguagens de programação propriamente ditas, os conceitos, os princípios e os modelos de programação e o estudo de estruturas de dados e de métodos de classificação e pesquisa de dados. Conforme consta na caracterização da matéria programação, entendida como programação de computadores, ela é uma atividade voltada à solução de problemas. Nesse sentido, ela está relacionada com uma variada gama de outras atividades como especificação, projeto, validação, modelagem e estruturação de programas e dados, utilizando as linguagens de programação como ferramentas.

Ao contrário do que se apregoava há alguns anos, a atividade de programação deixou de ser uma arte para se tornar uma ciência, envolvendo um conjunto de princípios, técnicas e formalismos que visam ao desenvolvimento de produtos de *software* bem estruturados e confiáveis. Cite-se, dentre estes, os princípios da abstração e do encapsulamento e as técnicas de modularização e de programação estruturada.

Portanto, o estudo de programação não se restringe ao estudo de linguagens de programação. As linguagens de programação constituem-se em uma ferramenta de concretização de produto de *software*, que representa o resultado da aplicação de uma série de conhecimentos que transformam a especificação da solução de um problema em um programa de computador que efetivamente resolve aquele problema.

No estudo de linguagens de programação, deve ser dada ênfase aos seus aspectos funcionais e estruturais, em detrimento aos detalhes de sintaxe. Conceitos como o significado de associação, avaliação, atribuição, chamada de procedimento, envio de mensagens, passagem de parâmetros, herança, polimorfismo e encapsulamento devem ser enfatizados. O estudo de linguagens de programação deve ser precedido do estudo dos principais paradigmas de programação, notadamente a programação imperativa, a funcional, a baseada em lógica e a orientada a objetos.

O desenvolvimento de algoritmos e o estudo de estruturas de dados devem receber especial atenção na abordagem do tema programação. Igualmente, deve ser dada ênfase ao estudo das técnicas de especificação, projeto e validação de programas. Um excelente campo para o exercício da programação é constituído pelo estudo de pesquisa em tabelas e de técnicas de ordenação (deve-se acrescentar o uso de estruturas de dados para o exercício da programação).

Dessa forma, deve-se entender que ensinar programação não é simplesmente ensinar uma linguagem de programação. Este ensino envolve sobretudo entender problemas e descrever formas de resolução, de maneira imparcial, para que então sejam codificadas em uma linguagem de programação. Ou seja, somente após o aprendizado dos conceitos de algoritmos e fundamentos de lógica, o estudante pode travar contato com uma linguagem de programação concreta para experimentar aqueles conceitos.

Além disso, o estudo de programação deve passar efetivamente pelo estudo de estruturas de dados. Assim, quando se projeta um programa, deve-se pensar nos algoritmos que manipulam os dados e na forma de estruturar estes dados. Desta maneira, as duas coisas se complementam.

De acordo com [1], no ensino de estruturas de dados, deve ser dado destaque à parte conceitual e comportamental das estruturas, antes de pensar em implementação. Por exemplo, no ensino de estruturas lineares, deve-se transmitir ao aluno o conceito desta estrutura, através das operações aplicáveis, como criar, acrescentar, consultar e remover elementos de uma lista encadeada. Em um curso de Ciência da Computação, esta parte pode ser apresentada através de uma especificação algébrica, por exemplo. Em seguida, discutem-se formas de implementações ou representações físicas das listas (contigüidade, encadeamento, circulares, duplamente encadeada, com/sem descritor, etc). Este tópico é onde normalmente o aluno apresenta maior dificuldade de assimilação do conceito, uma vez que o encadeamento não é um conceito muito freqüente na vida real, dificultando sua compreensão. Dessa forma, é recomendável que as disciplinas de estruturas de dados possuam significativa atividade prática em laboratório.

Com relação aos cursos na área de Computação e Informática, os cursos de licenciatura visam à formação

de educadores para o ensino médio. Os cursos de Ciência da Computação e Engenharia de Computação são voltados para a formação de recursos humanos para o desenvolvimento científico e tecnológico da computação. Embora conhecedores da matéria, não necessariamente devem sair dos cursos com o perfil de programadores. Os cursos de Sistemas de Informação incluem a formação de programadores entre seus objetivos. Nesse sentido, em um curso de Licenciatura em Computação, o aluno poderá ter uma carga de prática de programação menor do que a exigida nos demais cursos, sem prejuízo da parte conceitual da matéria. Deve haver uma diferenciação entre os casos de cursos de atividade fim e de atividade meio. Para este último, recomenda-se uma ênfase maior no conhecimento de linguagens de programação e atividades práticas.

Assim, a programação é, pois, um dos pontos-chaves em um curso de Computação, pois é a atividade que supre o computador com meios para servir ao usuário. Seu uso adequado e o entendimento de conceitos, princípios, teorias e tecnologias podem conduzir ao desenvolvimento de produtos de *software* de qualidade, objetivo final, tanto quando se tem a computação como uma atividade meio como quando a computação é tida como uma atividade fim.

Com esse intuito, [23] apresenta algumas recomendações genéricas a serem aplicadas no ensino de disciplinas relacionadas a algoritmos e computação:

- coerência com os objetivos fundamentais. Por coerência, entende-se que o professor deve: i) expressar claramente as idéias, os conceitos e as técnicas perante os alunos (se o professor coloca algoritmos confusos na lousa ou em transparências, ele não pode esperar algoritmos claros nas respostas dos alunos); ii) destacar a importância dos resultados teóricos e mostrar rigor formal toda vez que isto se fizer necessário; e iii) procurar valorizar o uso de técnicas na resolução de problemas. Esta última coerência pode ser alcançada em particular usando a técnica de descobrir a solução de um problema junto com os alunos, ao invés de simplesmente apresentar e explicar soluções prontas;
- ênfase no pensamento crítico. Este é papel de toda educação, porém, nesta matéria, um cuidado especial deve ser necessário dada sua natureza teórica com forte componente matemático. Em outras palavras, os alunos que têm pouca maturidade matemática tendem a acreditar em qualquer demonstração que lhes é apresentada. Tal comportamento deve ser desestimulado. É essencial que os alunos duvidem daquilo que lhes é apresentado e é com dúvidas saudáveis e sua resolução que a percepção da importância do resultado teórico poderá ser consolidada. Nesse sentido, considera-se um recurso valioso o conjunto de exercícios que pedem para os alunos identificarem falhas de argumentação (por

exemplo, em uma demonstração errada), erros em algoritmos ou erros em notícias da imprensa;

- a teoria na prática. A experiência com o ensino dessas disciplinas de Computação mostra que os alunos, em geral, não se sentem atraídos por elas, por considerarem-nas muito abstratas. Por esse motivo, crê-se ser importante usar como recurso didático sempre que possível um grande número de exemplos da vida real. A inclusão de projetos de implementação, seja dentro das disciplinas teóricas, seja dentro de uma disciplina específica, também visa a tornar a matéria menos abstrata. De resto, é importante salientar para os alunos o grande impacto que os resultados teóricos têm alcançado na prática.

3. Análise de Metodologias de Ensino de Computação abordando problemas e soluções

Dentro da esfera de discussão dos métodos educacionais como ferramentas para a formação de profissionais de qualidade para o mercado, o presente trabalho se propõe a dar sua contribuição. Entretanto, ele analisará um tema mais abrangente e complexo: as áreas de Computação e Informática, que apresentam inovações constantes e grande transformação de preceitos existentes. Sabe-se que é complicado promover mudanças na educação básica e, ao se tratar de cursos de graduação baseados na produção de tecnologia, o problema é ainda maior. O que se pode perceber é esses cursos exercerem grande atração sobre os futuros ingressos em uma faculdade, o que se mostra pela concorrência elevada em processos seletivos. No entanto, vencido o obstáculo de entrada, vêm o início da fase universitária e um dos maiores dilemas: o contato com disciplinas básicas que vão preparar o ingresso para atuar em áreas específicas. Esse contato, sobretudo em cursos de Computação, pode gerar tanto afinidade quanto repulsa, o que acontece com frequência expressiva, conforme [14].

Os problemas apresentados na compreensão de conceitos abstratos de programação, conforme [5], encontra sua raiz ainda durante a formação básica, o que caracteriza uma importante linha de pesquisa na metodologia educacional. Com certeza, sem uma radical mudança cultural de atitude e comportamento, um indivíduo dificilmente será um profissional de Computação adequado, visto que é preciso desenvolver a capacidade de buscar respostas através de novas perguntas que levem à resposta da questão original. A formação do profissional em Computação deve incitar o desenvolvimento de raciocínio crítico, a solução de problemas, a aplicação de métodos de pesquisa e o desenvolvimento profissional contínuo.

Dessa forma, a falta de compreensão do raciocínio lógico pode ser a principal razão pelo alto índice de reprovação nas disciplinas de Algoritmos e Programação e, em alguns casos, pela desistência de um curso. Em parte, isso é decorrência da dificuldade encontrada pelos

professores para acompanharem efetivamente as atividades laboratoriais de programação, dado o grande número de estudantes geralmente sob sua supervisão ([25]). Dentro desse paradigma, pode-se destacar a forma como os alunos estudam para a disciplina, geralmente memorizando, e a falta de pré-requisitos em conteúdos relacionados ([14]).

Em [19], são relatados outros problemas associados a este processo:

- a falta de motivação do aluno criada pelo despreparo, citado anteriormente, e o desânimo quando há, principalmente, na crença de que a disciplina constitui um obstáculo extremamente difícil de ser superado;
- o processo tradicional de avaliação pode deixar o aluno tenso prejudicando o aprendizado ([5]);
- o relacionamento entre professor e aluno pode ser um problema quando o primeiro preocupa-se em mostrar o que sabe desconsiderando um ambiente de aprendizagem descontraído e colaborativo;
- a didática ou a falta de metodologia de ensino dificulta o aprendizado dos novos e diversos conceitos. Inclui-se a grave falta de comunicação entre os professores das várias disciplinas que permitiria identificar conteúdos afins ou superposição de tópicos tornando o trabalho realizado mais integrado auxiliando o processo de ensino e aprendizagem ([12 e 17]).

Em [19], afirma-se que o professor deve ser capaz de fazer o aluno compreender a abstração envolvida com toda a simbologia utilizada. Para isto, deve utilizar sua criatividade e tentar resolver cada problema baseando-se em situações do cotidiano. Assim, o aluno começa a ter raciocínio lógico e ordem de pensamento.

Segundo [16], algumas das dificuldades vivenciadas pelo professor no processo são: i) reconhecer as habilidades inatas de seus alunos; ii) apresentar técnicas para soluções de problemas; iii) trabalhar a capacidade de abstração do aluno, tanto na busca das possíveis soluções como na escolha da estrutura de dados a ser utilizada; e iv) promover a cooperação e a colaboração entre os alunos.

No entanto, para [5], o iniciante do curso superior traz uma bagagem cultural que não apenas não serve para o apropriado desenvolvimento de um estudo acadêmico, mas pode atrapalhar o desenvolvimento desse estudo. Ele vem de uma experiência de estudante de disciplinas desenvolvidas em torno do paradigma de apresentação de respostas e soluções, usualmente da velha e surrada didática diretiva, ao invés de focar no processo de questionamento. Assim, mais uma das vertentes do problema educacional na área de Computação encontra-se na sua formação básica.

Mas, considerando o campo universitário, o nível de maturidade do aluno é importante neste ponto para que o professor possa adequar o processo pedagógico no sentido de acompanhar o amadurecimento do aluno,

permitindo aplicar técnicas de ensino e aprendizagem ([14]).

O acompanhamento contínuo de cada aprendiz e o tratamento personalizado a cada estudante, principalmente em turma com um número elevado de alunos, serão possíveis se forem apoiados por sistemas inteligentes, ou seja, auxiliados por computador ([17]). Logo, a idéia de usar produtos de *software* para o ensino de programação é de grande valia e ainda poderá transformar o ensino de disciplinas avançadas em algo mais primoroso e tecnológico.

A partir de todo esse problemático desafio, mudanças passaram a ser consideradas por graduados, mestres e doutores que trabalham com o ensino de algoritmos e programação. A análise da literatura feita por [14], independente de ser nacional ou internacional, mostrou que há três vertentes na busca de soluções para os problemas apontados:

- **Ferramentas**, que incluem trabalhos que apresentam ferramentas computacionais visando facilitar o processo de ensino e aprendizagem;
- **Estratégias**, que incluem trabalhos que discutem estratégias de ensino e/ou de avaliação de competências;
- **Ferramentas e Estratégias**, que são os trabalhos que discutem alguma estratégia suportada por ferramentas computacionais.

Juntamente com o uso de uma ferramenta computacional, deve-se tentar ajustar algumas propostas pedagógicas de trabalho disciplinar, cada vez mais foco de pesquisas na área de Computação, principalmente no âmbito internacional ([14]). Alguns trabalhos nacionais relacionados aos problemas de ensino de algoritmos e programação estão listados a seguir:

- em [8], percebendo os problemas e procurando solucioná-los, ou pelo menos amenizá-los, foi proposto o uso de uma abordagem de aprendizagem baseada na colaboração. Esta abordagem colaborativa parte do princípio de que o conflito sócio-cognitivo entre os aprendizes pode melhorar o seu aprendizado. Para suportar o aprendizado colaborativo, foi desenvolvido um avaliador de programas em C, chamado Verificador de Diferenças Significativas entre Programas – VDSP. O VDSP avalia os programas dos alunos, pontuando-os. A pontuação é utilizada para montar grupos de alunos que deverão refazer o programa, trocando pontos de vista e considerações. O VDSP foi experimentado com alunos do curso de Sistemas de Informação da PUC Minas Arcos, especificamente, alunos da disciplina Algoritmos e Estruturas de Dados, com resultados positivos em relação à formação de grupos de trabalho homogêneos e heterogêneos sob mediação do professor em detrimento de um trabalho individual;
- em [18], o aprendizado de algoritmos é considerado desafiador pelos estudantes de cursos tecnológicos. Os principais motivos para este fato são: o alto nível

de abstração do conteúdo e a dificuldade dos docentes em detectar e atender as dificuldades de aprendizagem dos alunos. Dessa forma, propuseram um ambiente que auxilia no atendimento às dificuldades de aprendizagem da disciplina de Algoritmos utilizando o paradigma de Assistentes Inteligentes de Ensino. Estes assistentes auxiliam o professor em sua tarefa de monitorar, assistir e analisar os alunos. Os resultados da utilização do ambiente apontam para a viabilização da tarefa de acompanhamento individualizado dos alunos e uma maior eficiência na identificação e atendimento às dificuldades de aprendizagem;

- em [4], discute-se que grupos de estudo têm surgido como uma alternativa para complementar os conteúdos vistos em sala de aula e desenvolver/aprimorar habilidades. Neste contexto, apresentaram a importância dos grupos de estudo na formação dos futuros profissionais da área de computação. Relataram a experiência com o grupo de estudos de Java da Universidade Luterana do Brasil (ULBRA). Ao contrário do que se possa imaginar, a diferença de nível entre os alunos não constituiu obstáculos para o andamento dos trabalhos. Observou-se uma forte interação entre os componentes do grupo e o prazer em dividir o conhecimento adquirido. Relações se estabeleceram a partir da experiência de pesquisar, aprender e resolver problemas juntos;
- em [9], foram concluídos, após diversos estudos sobre Ambientes de Aprendizado Colaborativo Suportado por Computador, que eles podem trazer melhorias ao processo de aprendizado e contribuir para torná-lo mais prazeroso e socializador. Então, desenvolveram um produto de *software* verificador de diferenças significativas entre programas e o utilizaram para a montagem de grupos para o aprendizado colaborativo de programação de computadores. Chegaram à conclusão de que o aprendizado colaborativo é viável e eficaz nesse contexto, mas difícil de implantar sem a presença de um mediador. Logo, estudaram o *Learn in Group*, um sistema baseado na *Web* para o suporte ao aprendizado colaborativo de programação. O sistema conta com dois agentes mediadores, um para a constituição de grupos de trabalho colaborativo e o outro para acompanhar e estimular a participação dos membros dos grupos;
- em [15], foi citado que o Grupo de Trabalho de Licenciatura em Computação (GT-3) da Sociedade Brasileira de Computação (SBC) estabeleceu, entre as metas para 2004-2005, desenvolver experiências relativas ao ensino de programação no nível médio. Isso visando melhorar o comportamento dos futuros ingressos em cursos de Computação e Informática, tentando sanar os problemas relacionados ao pensamento algorítmico mais cedo. A partir dessa idéia, pesquisadores da área da Universidade

Estadual Norte Fluminense (UENF) idealizaram uma proposta para ser aplicada em uma escola pública de Cardoso Moreira – RJ, através de uma oficina que permitiu identificar os pontos de dificuldade no processo e a necessidade de trabalhar mais a informática de modo interdisciplinar. Apesar da elevada evasão, o resultado obtido na avaliação dos concluintes da oficina permite projetar as expectativas da inclusão do tema algoritmos e programação na matriz curricular. A oficina permitiu comprovar que a idéia do GT-3 é factível, ou seja, que alunos do ensino médio são capazes de desenvolver tal conteúdo sem necessariamente estarem inseridos em um curso técnico, o que pode estimular o professor por aproveitar a nova competência do aluno para promover o aprofundamento de conteúdos de sua disciplina e, conseqüentemente, a interdisciplinaridade;

- em [27], foi discutida a experiência do uso de projetos integrados para promover a interdisciplinaridade no curso de Ciência da Computação. A integração tanto ocorre entre disciplinas de um mesmo período quanto entre disciplinas de diferentes períodos. Esta última integração foi altamente favorável entre as disciplinas que formam os pacotes de disciplinas optativas, ofertadas a partir do sexto semestre. Cada projeto visou transpor as barreiras da fragmentação do conhecimento (não se trata de uma colagem fragmentada de saberes), o aluno foi posto diante de problemas que exigem a aplicação dos conhecimentos de uma disciplina em outra, para posteriormente utilizá-los na resolução dos problemas. Esse processo visou estabelecer o valor das competências e habilidades em converter o conhecimento específico da área de formação para outras áreas de estudo. Os resultados mais expressivos se refletiram no comportamento dos alunos: criou-se uma cultura do ensino-aprendizado pelo próprio esforço, de que o processo de aprendizado deve ser, em grande parte, construído pelo próprio aluno. Para os docentes, ficou claro que devem estar aptos a aceitarem novos papéis e ficarem atentos ao processo dinâmico que é o ensino da Ciência da Computação. Os projetos iniciaram uma série de discussões sobre a fragmentação das disciplinas envolvidas nos projetos. Foram exibidas inúmeras deficiências de disciplinas dos períodos anteriores ao quarto período. Foi proposto um questionamento dos conteúdos programáticos de diversas disciplinas e suas formas de abordagens. Finalmente, foram relevantes nas discussões sobre as novas propostas para uma total reestruturação curricular do curso de Bacharelado em Ciência da Computação, visando melhorar a metodologia de ensino;
- em [13], afirmou-se que os programas educacionais, conforme a metodologia utilizada pelo professor,

podem ser explorados de diferentes formas. Neste sentido, estes ambientes requerem um conjunto de funções para atender os aspectos pedagógicos, identificados pelos professores especialistas. Como esta tarefa não é de fácil execução, dado o perfil diferenciado da equipe interdisciplinar, fez-se necessário a utilização de uma metodologia para guiar e organizar todo o projeto de desenvolvimento do produto de *software*. Dessa forma, eles apresentaram um relato de experiência em uma disciplina da graduação da Pontifícia Universidade Católica do Rio Grande do Sul (PUCRS) no intuito de formalizar as práticas de Engenharia de *Software* adotadas para o projeto e desenvolvimento de produtos de *software* educacionais, através do projeto PDS-E (Processo para Desenvolvimento de *Software* Educacional). Isso para que sistemas de informação educacionais fossem desenvolvidos de forma estruturada e pedagógica, contendo o assunto a ser tratado. Os resultados obtidos com a aplicação da disciplina em três semestres consecutivos foram muito promissores. Os alunos, em seus depoimentos colhidos através da avaliação formal e no decorrer da disciplina, afirmaram que desta maneira conseguiram dimensionar melhor o trabalho a ser feito e conversar melhor com os programadores. Os programadores, por sua vez, entendem melhor o que deveriam fazer. Enfim, todos ganharam com a formalização e a padronização das idéias. Neste contexto, tem-se buscado formalizar a metodologia de ensino relatada e o uso dos recursos através da proposta do PDS-E.

Para [11], diversos sistemas para implementação de animações de algoritmos e de estruturas de dados foram produzidos. Vários destes sistemas exploram a potencialidade do uso de visualizações gráficas das operações realizadas nas estruturas de dados como ferramenta de ensino. O uso de movimento em tempo real, cores e sons enriquecem ainda mais o poder de comunicação. De fato, algumas experiências positivas foram relatadas na literatura, principalmente com o uso de sistemas que utilizam a abordagem ativa, na qual o aprendiz interage com as animações na criação de instâncias para testes. Por outro lado, a ênfase na qualidade da animação gráfica para enriquecer a abordagem ativa da maioria dos sistemas levou-os a um grau de complexidade que impede seu uso para fins de criação de animações pelos próprios aprendizes. Deste modo, sua utilização se limita à interação, ainda que ativa, com as animações pré-implementadas, representando as mais novas tendências de trabalhos para o ensino de Computação e Informática.

Em [11], afirmou-se que, do mesmo modo que a experimentação durante a operação das animações enriquece o aprendizado mais do que a mera observação passiva delas, é de se esperar que, com um sistema onde a própria implementação das animações gráficas é facilitada a ponto de poder ser realizada pelo estudante,

a absorção do funcionamento dos algoritmos seja ainda mais intensa.

Apesar desta nova tendência dos produtos de *software* para educação em Computação, ainda é muito utilizada a abordagem ativa descrita anteriormente. Isso porque configura uma forma de controle do aprendizado pelo discente, uma vez que quaisquer informações erroneamente assimiladas na maioria das vezes causam um impacto negativo na evolução do aluno durante o curso. Dessa forma, o produto de *software* utiliza estratégias passo a passo na simulação de estruturas de dados conhecidas e bem sedimentadas. Assim, o aluno realiza uma série de operações ao utilizar o programa, assimilando conceitos básicos sem correr o risco de implementar seus próprios algoritmos de forma incorreta.

4. A importância do uso de ferramentas computacionais no ensino

É importante salientar alguns pontos especiais:

- a aula de uma disciplina que relaciona programação, algoritmos e estruturas de dados deve ser realizada em laboratórios para que os alunos possam entender as abstrações apresentadas. Deve ser composta de uma parte teórica, onde conceitos são transmitidos, e uma parte prática, onde são entendidos comandos e tópicos ministrados. Deve haver questões-desafio, para que os discentes sejam acostumados a usar de agilidade e criatividade na solução de problemas ([14]);
- o uso de um produto de *software* educacional, que enfatize animação gráfica, é de extrema importância como facilitador do processo de aprendizagem, visto que a apresentação de conceitos abstratos se torna mais viável e didática, melhorando a qualidade do material das aulas ([24]);
- normalmente, um aluno se interessa por aulas diferenciadas e isso não só prende sua atenção, como também influenciam positivamente nas avaliações ([5]);
- uma economia de tempo pode ser conseguida e seria direcionada para maiores explicações e resoluções de exercícios, uma vez que o material didático seria virtual, evitando uma explanação cansativa através do quadro-negro ([20]);
- o uso de uma boa ferramenta que seja livre permite ao aluno ter acesso ao material virtual para estudar em casa. Além disso, ele possuiria um valioso projeto, para consultas futuras, em caso de necessidade – a tendência em abstrair detalhes, com o passar do tempo, sempre leva a dúvidas e estas a novas pesquisas em estruturas básicas ora aprendidas ([11]);
- a existência de um material também via *web* para estudos e pesquisas estimula maior contribuição tecnológica para a sociedade, integração de diferentes ideais através de contatos pela *Internet* e

maior alcance ao público. Além disso, a divulgação das propostas de melhorias de processos educacionais visa injetar e obter informações para ampliação da literatura disponível ([21]);

- a transição de educadores seria facilitada, uma vez que existiria uma base pronta para ser apreciada, mantendo a qualidade de ensino e aprendizagem do conteúdo.

Assim, o uso de um sistema de informação educacional, o produto de *software* educacional, é um subsídio valioso para a formação de alunos, ajudando-os a concluírem seu curso, principalmente se atender a disciplinas que estão no início do curso e são pré-requisitos necessários para as disciplinas mais específicas de períodos avançados. Quando feita uma boa base, rendimento e desempenho aumentam, proporcionando melhores resultados, melhores currículos e melhores profissionais para o mercado ([20]).

5. Ferramentas para o ensino de algoritmos, estruturas de dados e programação

Para o desenvolvimento de qualquer trabalho, faz-se necessário discutir o que realmente há de contribuição ao universo acadêmico. Nesta seção, são citados alguns projetos relacionados. Deve-se ressaltar, contudo, que não houve pretensão de fazer uma análise exaustiva de tais projetos. São eles:

- *ASTRAL (Animation of Data Structures and Algorithms)* [11]. É um ambiente de programação para produção de animações de algoritmos e de estruturas de dados com propósito instrucional. O ambiente foi desenvolvido no Instituto de Computação da Universidade de Campinas (IC/UNICAMP) em 1995-97 para plataforma Macintosh e recentemente portado para Windows. Utilizando o ambiente, um usuário pode implementar uma variedade de estruturas de dados e algoritmos (que atuam sobre essas estruturas) e, através de chamadas de algumas poucas rotinas específicas do ambiente, é possível visualizar graficamente as estruturas de dados bem como animações que mostram a atuação dos algoritmos sobre elas. Isso se dá através de uma biblioteca de suporte gráfico que é parte do *ASTRAL*. A abordagem de animação de algoritmos tem duas vantagens imediatas: aprendizado e depuração. Aquele se beneficia da observação dinâmica do funcionamento das estruturas de dados e proporciona um nível de abstração maior, enquanto a depuração é facilitada pela visualização gráfica que deixa claro para o usuário-programador, no momento da execução, em que parte da implementação da estrutura ou do algoritmo os erros ocorrem. Ambas vantagens são particularmente interessantes para alunos de disciplinas de estruturas de dados e algoritmos. O uso do *ASTRAL* em disciplinas de graduação sobre

estruturas de dados como componente de programação foi muito bem sucedido durante vários semestres;

- EDDL (Estruturas de Dados Dinâmicas Lineares) [2]. Surgiu de uma discussão sobre estratégias de ensino das estruturas de dados e tipos de dados abstratos utilizados em programação. A estruturação do programa EDDL e a concepção da sua funcionalidade englobam os seguintes componentes: i) uma abordagem introdutória aos conceitos ou especificação dos tipos de estruturas; ii) as técnicas de implementação; e iii) a utilização dessas estruturas na resolução de problemas. Pode ser utilizado quer como ferramenta introdutória como ferramenta de revisão, exercitação e auto-avaliação. Após o desenvolvimento dos primeiros protótipos ou versões experimentais, o EDDL adquiriu uma configuração global em que uma tela principal dá acesso aos módulos principais do programa, que são os seguintes: conceitos introdutórios, pilhas, filas, listas lineares e exercícios. Em cada uma dessas seções, é analisada em detalhe cada uma das operações mais usuais com estes tipos de estruturas, tais como: criação e destruição da estrutura, inserção e remoção de elementos na estrutura e verificação do estado da estrutura (vazia, cheia). Oferece um conjunto de ferramentas e outros recursos de programação que o revelam ser ao mesmo tempo intuitivo e de fácil manipulação, diversificado, versátil e eficaz para o desenvolvimento de projetos, principalmente quando o aspecto gráfico é uma das componentes importantes a ter em conta;
- TED (Tutorial de Estruturas de Dados) [10]. É um tutorial para dinamizar o ensino na disciplina de estrutura de dados. O tutorial desenvolvido é um produto de *software* educacional com o objetivo de permitir uma maior interação entre o professor e o aluno, servindo como uma ferramenta de apoio ao aprendizado. O tutorial subdivide-se nas várias estruturas apresentadas durante a disciplina, permitindo ao aluno ter uma visão individualizada dos conteúdos, mas sem perder a visão do todo. Através de conceitos de dinamização de aprendizagem, como a visualização dos acontecimentos simultaneamente com o que é executado, o aluno cria uma visão de totalidade do sistema. O sistema de tutorial apresenta os algoritmos das estruturas de dados, ao mesmo tempo em que demonstra a sua implementação lógica através de exemplos visuais permitindo ao aluno acompanhar todos os passos que são executados;
- *Dictionary of Algorithms and Data Structures* [3]. Este *site* contém uma parte do *Software Quality Group* do *Software Diagnostics and Conformance Testing Division, Information Technology Laboratory*. Este é um dicionário de algoritmos, técnicas algorítmicas, estruturas de dados, problemas atípicos e definições relativas. Os algoritmos

incluem funções comuns e os problemas incluem alguns como o caixeiro viajante. Alguns tópicos têm *links* para implementações e mais informações. A página de índice relaciona os tópicos por tipo ou em ordem alfabética. Apresenta o funcionamento dos algoritmos como animações sendo executadas ao se definirem os seus parâmetros;

- TBC-AED (Treinamento Baseado em Computador para Algoritmos e Estruturas de Dados) [20 e 22]. É um produto de *software* educacional que analisa tópicos básicos de programação, englobando um conteúdo teórico sintético e direto, acompanhado de processo gráfico passo a passo, representando um repositório didático com visualização gráfica. Isso facilita a visualização e o entendimento das informações ora apresentadas e disponibiliza o tempo que seria gasto em transcrever explicações do quadro-negro para o caderno para a resolução de exercícios de aplicação e fixação. Além disso, pode-se alcançar maior interação entre o professor e os seus alunos, no sentido de aumentar o espaço para questionamentos. Possui algumas características, tais como: i) *links* explicativos, evitando a necessidade de aprendizagem via tutorial; ii) a usabilidade da interface gráfica é razoável, possibilitando ao professor apresentar conceitos iniciais, conteúdos teóricos e práticos aos poucos, como apresentaria em transparências; iii) conteúdo teórico simples, de forma a familiarizar melhor o aluno com o assunto; iv) processo gráfico passo a passo, com elementos numéricos, o que melhora a visualização e o entendimento; e v) legendas explicativas, que ilustram as etapas do processo de apresentação de algoritmos, facilitando o aprendizado e englobando aulas teóricas e práticas, ao envolver uma nova forma de transmitir conhecimento. O TBC-AED apresenta os seguintes temas: Busca Binária, Métodos de Ordenação (*Select Sort*, *Insert Sort*, *Bubble Sort*, *Merge Sort* e *Quick Sort*), Estruturas de Alocação Estática e Dinâmica de Memória (Lista, Fila e Pilha) e Árvore Binária de Busca. Ele foi elaborado usando Java e o ambiente de desenvolvimento *NetBeans*. A partir da apresentação e disponibilização da ferramenta aos alunos, aqueles que a utilizaram obtiveram êxito na disciplina, perfazendo um total de aprovação superior àqueles que não a utilizaram. Devido a esse fato, para melhorar os resultados, optou-se por dividir a carga horária da disciplina, antes apenas teórica, em aulas expositivas (teóricas) e de laboratório (uso da ferramenta, incluindo exercícios de programação) para o segundo semestre de 2005. Como consequência da evolução, intrínseca a qualquer projeto, o TBC-AED foi disponibilizado via *web*, através de *applets*¹, gerando o TBC-AED/WEB

¹ *Applets* são programas Java que podem ser embutidos em documentos HTML. Quando um navegador carrega

(Treinamento Baseado em Computador para Algoritmos e Estruturas de Dados via *Web*) para que possa ter maior acessibilidade, além de possibilitar a troca de idéias, através da interatividade com usuários diversos ([21]). A idéia de criar o TBC-AED/WEB surgiu para evitar que o estudante precise fazer o *download* do TBC-AED para estudar, bastando ter uma versão do aplicativo apenas em sua estação de trabalho em casa, para trabalhar *offline*. Quando estiver em alguma estação conectada à *Internet*, ele pode executar o programa diretamente no navegador, desde que esta estação possua a JVM (*Java Virtual Machine*) instalada. Quanto às características relacionadas à funcionalidade e à aparência, o TBC-AED/WEB não difere do TBC-AED. Dessa forma, a partir do acesso *web*, espera-se gerar uma interatividade maior com outros pesquisadores, difusão da pesquisa, contribuição à comunidade científica da área e *feedback* em relação ao trabalho realizado.

6. Conclusões

Sob todo conjunto de informações apresentadas até o presente momento, pode-se perceber que existe uma gama de recursos que possibilitam o desenvolvimento de ferramentas computacionais para o auxílio da educação superior em Computação, com destaque para a linguagem de programação Java, que conta com a portabilidade e forte suporte para técnicas adequadas de Engenharia de *Software* ([7]). Isso faz com que as novas tecnologias atinjam mais rapidamente seu público alvo.

Além disso, observou-se que o uso de produtos de *software* para o ensino de programação e algoritmos é uma idéia inteligente, se for amadurecida e desenvolvida de forma cuidadosa e estruturada. Isso fornece novas experiências para professores e alunos, que se proponham a trabalhar com esse tema. Além disso, eles terão sua qualidade de ensino elevada e poderão ser capazes de avançar mais rapidamente no campo do conhecimento, através da agilidade de processos didáticos e da participação ativa do docente da área.

Conclui-se também que uma boa elaboração de produtos tecnológicos que facilitem a transmissão de conhecimentos em Computação deve ser acompanhada de expressiva pesquisa no campo de novas metodologias de ensino. A principal finalidade disso é, cada vez mais, haver contribuições que incorram no aprimoramento do ensino superior e na formação de profissionais melhor qualificados para o mercado. Isso reflete principalmente sobre futuros professores, uma vez que, aqueles que têm passado por esse tipo de experiência durante a graduação, terão grande interesse em executar o mesmo processo quando estiverem lecionando.

uma página da *Web* que contém um *applet*, o *applet* é executado ([7]).

Dessa forma, compreende-se que a educação precisa ser reavaliada, pois representa a base da atual sociedade mundial e o único instrumento que pode desenvolver o ser humano e torná-lo digno de constituir sua vida e seu ambiente. Isso deve ser acompanhado por uma atitude crítica da comunidade científica formada, que é a maior interessada em manter e prosseguir com avanços que contribuam para a melhoria da qualidade de vida.

7. Referências

- [1] Azeredo, P. A. *Uma proposta de Plano Pedagógico para a Matéria de Programação*. Anais do II Curso: Qualidade de Cursos de Graduação da Área de Computação e Informática (WEI 2000). Editora Universitária Champagnat.
- [2] Azul, A. A. e Mendes, A. J. *EDDL: Um Programa Didático sobre Estruturas de Dados Dinâmicas Lineares*. 3º Simpósio Investigação e Desenvolvimento de Software Educativo – 1998. Évora, Portugal. Disponível em: <http://www.minerva.uevora.pt/simposio/comunicacoes/Mendes/Eddl.html>. Acesso: 15 jun 2005.
- [3] Black, P. E. *Dictionary of Algorithms and Data Structures*. NIST (National Institute of Standards and Technology). Disponível em <http://www.nist.gov/dads/>. Acesso: 15 jun 2005.
- [4] Borges, K. S. e Filho, H. B. R. *A Importância dos Grupos de Estudos na Formação Acadêmica*. XIII Workshop de Educação em Computação (WEI'2005). São Leopoldo, RS, Brasil.
- [5] Buzin, P. F. W. K. *A epistemologia da Ciência da Computação: Desafio do Ensino da Ciência*. Revista de Educação, Ciência e Cultura, v. 6, n° 2. Centro Universitário La Salle. Canoas, RS, Brasil, 2001.
- [6] Campos, E. A. V. e Ascencio, A. F. G. *Fundamentos da Programação de Computadores*. Editora Prentice Hall, 2003.
- [7] Deitel, H. M. e Deitel, P. J. *Java Como Programar*. Bookman Companhia ed Informática, 2003.
- [8] Faria, E. S. J. e Coello, J. M. A. *Um Estudo Empírico dos Efeitos do Uso de Trabalho Colaborativo no Aprendizado de Programação em Cursos de Graduação em Computação*. XIII Workshop de Educação em Computação (WEI'2005). São Leopoldo, RS, Brasil.
- [9] Faria, E. S. J., Vilela, J. M. e Coello, J. M. A. *Um Sistema de Aprendizado Colaborativo de Programação Baseado em Agentes chamado Learn In Group*. XIII Workshop de Educação em Computação (WEI'2005). São Leopoldo, RS, Brasil.
- [10] Flávio, D. *TED - Tutorial de Estruturas de Dados*, desenvolvido durante estágio supervisionado na Universidade do Vale do Itajaí – 2004. Disponível

- em <http://www.tutorialdeestruturadedados.8m.com>. Acesso: 20 jun 2005.
- [11] Garcia, I. C., Rezende, P. J. e Calheiros, F. C. *Astral: Um Ambiente para Ensino de Estruturas de Dados através de Animações de Algoritmos*. Revista Brasileira de Informática na Educação (RBIE'1997) nº 01, <http://www.inf.ufsc.br/sbcie/revista/nr1/garcia.htm>. Ambiente ASTRAL disponível em: <http://www.dcc.unicamp.br/~rezende/ASTRAL/>. Acesso: 28 abr 2005.
- [12] Giraffa, L., Marczak, S. e Almeida, G. *O Ensino de Algoritmos e Programação mediado por uma Ambiente Web*. Congresso Nacional da Sociedade Brasileira de Computação (SBC'2003). Campinas, SP, Brasil.
- [13] Giraffa, L., Marczak, S. e Prikkladnicki, R. *PSD-E: Em Direção a um Processo para Desenvolvimento de Software Educacional*. XI Workshop de Informática na Escola (WIE'2005). São Leopoldo, RS, Brasil.
- [14] Júnior, J. C. R. P. e Rapkiewicz, C. E. *O Processo de Ensino e Aprendizagem de Algoritmos e Programação: Uma Visão Crítica da Literatura*. III Workshop de Educação em Computação e Informática do estado de Minas Gerais (WEIMIG'2004). Belo Horizonte, MG, Brasil.
- [15] Júnior, J. C. R. P., Rapkiewicz, C. E., Delgado, C. e Xexeo, J. A. M. *Ensino de Algoritmos e Programação: Uma Experiência no Nível Médio*. XIII Workshop de Educação em Computação (WEI'2005). São Leopoldo, RS, Brasil.
- [16] Menezes, C. e Nobre, I. *Suporte à Cooperação em um Ambiente de Aprendizagem para Programação (Samba)*. XIII Simpósio Brasileiro de Informática na Educação (SBIE'2002). São Leopoldo, RS, Brasil.
- [17] Pimentel, E. P., França, V. F. e Omar, N. *A Caminho de um Ambiente de Avaliação e Acompanhamento Contínuo de Aprendizagem em Programação de Computadores*. II Workshop de Educação em Computação e Informática do Estado de Minas Gerais (WEIMIG'2003). Poços de Caldas, MG, Brasil.
- [18] Raabe, A. L. A. e Silva, J. M. C. *Um Ambiente para Atendimento das Dificuldades de Aprendizagem de Algoritmos*. XIII Workshop de Educação em Computação (WEI'2005). São Leopoldo, RS, Brasil.
- [19] Rodrigues, M. C. *Como Ensinar Programação?*. Informática – Boletim Informativo Ano I nº 01, ULBRA. Canoas, RS, Brasil, 2002.
- [20] Santos, R. P. e Costa, H. A. X. *TBC-AED: Um Software Gráfico para Apresentação de Algoritmos e Estruturas de Dados aos Iniciantes em Computação e Informática*. I Congresso de Computação do Sul do Mato Grosso (COMPSULMT'2005). Rondonópolis, MT, Brasil.
- [21] Santos, R. P. e Costa, H. A. X. *TBC-AED e TBC-AED/WEB: Um Desafio no Ensino de Algoritmos, Estruturas de Dados e Programação*. IV Workshop de Educação em Computação e Informática do estado de Minas Gerais (WEIMIG'2005). Varginha, MG, Brasil.
- [22] Santos, R. P. e Costa, H. A. X. *Desenvolvimento de um Software Educacional para o Ensino de Algoritmos, Estruturas de Dados e Programação*. CD de Resumos XVIII Cicesal – XII Seminário PIBIC/Cnpq – VIII Seminário PBICT/Fapemig - I Congresso de Extensão, Lavras, MG, v. 1, p. 113-113, 2005.
- [23] Setubal, J. C. *Uma proposta de Plano Pedagógico para a Matéria de Computação e Algoritmos*. Anais do II Curso: Qualidade de Cursos de Graduação da Área de Computação e Informática (WEI 2000). Editora Universitária Champagnat.
- [24] Soares, T. C. A. P., Cordeiro E. S., Stefani Í. G. A., Tirelo, F. *Uma Proposta Metodológica para o Aprendizado de Algoritmos em Grafos Via Animação Não-Intrusiva de Algoritmos*. III Workshop de Educação em Computação e Informática do Estado de Minas Gerais (WEIMIG'2004). Belo Horizonte, MG, Brasil.
- [25] Tobar, C. M., Rosa, J. L. G., Coelho, J. M. A. e Pannain, R. *Uma Arquitetura de Ambiente Colaborativo para o Aprendizado de Programação*. XII Simpósio Brasileiro de Informática na Educação (SBIE'2001). Vitória, ES, Brasil.
- [26] Wirth, N. *Algoritmos e Estruturas de Dados*. LTC Informática-Programação, 1989.
- [27] Yamamoto, F. S., Silva, A. F., Zanutto, J. e Zampirolli, F. A. *Interdisciplinaridade no Ensino de Computação*. XIII Workshop de Educação em Computação (WEI'2005). São Leopoldo, RS, Brasil.