

Em Busca de uma Prática Pedagógica Inclusiva: Uma Experiência no Ensino de Estruturas de Dados Mediada por Animações Gráficas

Benedito J. P. Ferreira

¹Departamento de Informática - Universidade Federal do Pará
R. Augusto Correa, 01 – CCEN - 66.075-110 - Belém, PA, Brasil

Resumo. *Este artigo descreve uma experiência realizada na disciplina Estrutura de Dados, buscando-se maior eficácia do processo de ensino-aprendizagem, por meio de um processo avaliativo que alia diagnóstico e ação transformadora de forma indissociável. Nesse contexto, animações gráficas são empregadas como recurso pedagógico. Conclui-se com indicações do caráter promissor dessa estratégia, tendo-se em vista os resultados alcançados.*

Abstract. *This paper reports an experience that aimed to increase the efficacy of the teaching-learning process. This experience was conducted in a Data Structures class using and used an evaluation process that applies diagnostic and corrective actions in combination. In this experience, graphic animations are used as pedagogical concepts. The results of this experience suggest that this is a promising line of action to be explored.*

1. Introdução

O ensino/aprendizagem de programação tem merecido especial atenção nos últimos anos, tanto em termos de discussão de questões metodológicas quanto de proposição de ferramentas ou novos conceitos relacionados¹. O que explica esse interesse é a complexidade de tal processo, que envolve muito mais do que simples transferência/aquisições de informações. A aprendizagem efetiva de programação exige do educador a concepção de processos onde o aluno possa desenvolver uma específica capacidade de abstração: a de mobilizar um conjunto de ferramentas simbólicas, que representam ações computacionais, para a resolução de problemas de natureza variada. Em outras palavras, é necessário guiar o futuro programador em um processo de capacitação para *conceber* ou *criar* uma solução a partir de um ferramental lógico/simbólico disponível.

A disciplina de Estrutura de Dados, que em geral, nos currículos dos cursos de computação, sucede a de programação básica, cumpre um papel fundamental. Primeiramente porque, situando-se no segundo ou terceiro semestre dos cursos, deve oferecer o amadurecimento necessário das técnicas fundamentais já aprendidas de programação imperativa. Além disso, fornecerá novas técnicas (e aprofundará as já conhecidas) de estruturação dos dados, o que será fundamental em toda a prática futura de programação do educando. Entretanto, observa-se a ocorrência de elevados índices

¹ Cf. Kelleher 2005, Byckling e Sajaniemi, 2006, Olsen 2005, Ferreira, 2006, dentre outros.

de evasão, reprovação e de baixo rendimento² nessa disciplina, ocasionando-se uma série de problemas tanto para o aluno (em disciplinas subseqüentes e em sua futura prática profissional) como para a gestão de um curso de Computação, dado o “represamento” de alunos demandando a disciplina. Se esse problema não for adequadamente tratado, a tendência é de agravamento (por exemplo, a oferta de turmas muito grandes para atender a essa demanda).

Este artigo relata uma experiência na condução de uma turma de Estrutura de Dados na Universidade Federal do Pará, no segundo semestre de 2006, visando-se a uma prática pedagógica *inclusiva*, aqui entendida como a que seja capaz de proporcionar condições para que o maior número possível de alunos atinja os objetivos estabelecidos. Buscou-se essa meta através de um processo avaliativo acoplado à adoção de ações corretivas, adaptadas tanto quanto possível ao ritmo de cada educando. Na seção 2 serão apresentadas as principais referências conceituais dessa experiência. Na seção 3, será discutido o potencial pedagógico do uso de animações gráficas no ensino de programação, particularmente de estruturas de dados. Na seção 4, será descrita em detalhes a experiência realizada, com os resultados alcançados. Na seção 5, são feitas as considerações finais, centrando-se no potencial dos princípios e procedimentos adotados para outras experiências semelhantes.

2. Pressupostos Conceituais e Metodológicos

Nesta seção, apresenta-se a base conceitual da experiência descrita na seção 4. Inicialmente, explicita-se a distinção, nem sempre clara, entre avaliação e aplicação de exames, bem como a importância dessa concepção para a uma prática educativa inclusiva. Na seqüência, busca-se reunir princípios metodológicos que possam, após feito o diagnóstico, orientar as ações a serem adotadas no sentido de se atingirem os objetivos estabelecidos.

2.1. Diagnóstico pede ação, ação pressupõe diagnóstico

Luckesi (2005) afirma que, embora a expressão “avaliação da aprendizagem” seja bastante consagrada no meio educacional, a prática disseminada nesse meio, desde o ensino infantil até a pós-graduação, é a prática de exames. O principal argumento que dá sustentação a essa afirmação é o de que as provas ou exames aplicados rotineiramente acabam assumindo muito mais o papel de medir o grau de assimilação de informações ou aquisição de habilidades, objetivando à atribuição de uma nota ou um conceito (certificação da aprendizagem).

Para o autor, a adoção de exames *por si* somente faz sentido quando se tem efetivamente a necessidade de medir conhecimentos e habilidades, como é o caso de concursos, certificações ligadas a associações profissionais, etc. Entretanto, no processo educativo, o objetivo primordial não é medir conhecimentos, mas promover o crescimento do educando em etapas sucessivas e de acordo com objetivos claramente estabelecidos. Portanto a avaliação deve ser sempre subsidiária desse objetivo maior, e

² Baixo rendimento é entendido aqui como obtenção de nota mínima 5, em uma escala de 0 a 10, ou pouco acima, o que corresponderá, em um sistema de conceitos, ao “regular”. Os indicadores de insucesso foram observados especialmente no registro histórico de turmas ofertadas em anos recentes na Universidade Federal do Pará e ainda em contatos com coordenadores de cursos de outras instituições.

nessa perspectiva, diagnóstico e ação precisam andar sempre juntos; assim como não faz sentido agir sem fazer diagnósticos, que forneçam indicações do bom andamento (ou não) do processo educativo em curso, igualmente não faz sentido diagnosticar sem a subsequente correção de rumos, tendo-se em vista que o objetivo estabelecido seja alcançado. Essa é a verdadeira avaliação (Idem: 34).

No sentido acima expresso, exames são classificatórios e excludentes; já a avaliação é inclusiva e acolhedora. Aceita o educando com seu potencial e suas limitações, buscando facilitar-lhe o caminho para seu crescimento.

2.2. Orientações para a ação

Vários estudos têm procurado identificar as habilidades necessárias à tarefa de programação, o que, conseqüentemente pode fazer indicações acerca das dificuldades no aprendizado desse conteúdo e propor elementos para uma metodologia que supere esses obstáculos. Ferreira (2006) apresenta um conjunto de princípios metodológicos, que são sintetizados abaixo:

- **Progressividade:** a aquisição de boa desenvoltura em um conjunto de ferramentas fundamentais é condição necessária para que ferramentas mais avançadas sejam apresentadas;
- **Ritmo individual mediado por avaliação:** devem ser definidos níveis de complexidade, e a passagem para um novo nível só deve se dar quando se avaliar satisfatoriamente a desenvoltura do aluno com os conceitos do nível anterior.
- **Centralidade na resolução de problemas:** deve-se evitar que detalhes de sintaxe ou outros fatores (conceitos apresentados prematuramente como os relacionados com programação modular ou orientada a objetos) funcionem como elemento dificultador do processo de aprendizagem de resolução de problemas. Propõe-se o uso de um conjunto de *cartelas* com a descrição sintática dos principais recursos da linguagem adotada, necessários em determinado *nível*.
- **Prática laboratorial permanente:** como o aprendizado de programação envolve o desenvolvimento de uma habilidade (mobilizar ferramentas simbólicas para a resolução de um problema), ele não será bem sucedido sem a participação ativa do aluno na resolução de problemas, momento em que ocorrerá a apreensão do significado concreto de cada elemento simbólico presente na sua proposta de solução. É o momento também em que as dificuldades afloram, o que requer assistência constante.
- **Aprendizagem com trocas mútuas:** em determinadas atividades, pode-se propor o agrupamento de alunos com diferentes níveis de amadurecimento, para que haja troca de informações entre eles. Assim, explora-se um conhecido princípio educacional de que o ensino, ao exigir a sistematização do que é ensinado, é uma excelente oportunidade de aprendizado.

3. Técnicas de Animação no Desenvolvimento da Capacidade de Abstração

Um dos aspectos fundamentais da tarefa de programação é a abstração. Um comando ou expressão em linguagem de alto nível é uma representação simbólica de um conjunto de ações concretas de tratamento dos dados. Assim, o domínio dessas abstrações, ou seja, a

percepção do significado efetivo de cada elemento simbólico do programa é fundamental para a aprendizagem de programação. A percepção do significado das diversas estruturas de dados empregadas e das transformações que os valores nelas armazenados vão sofrendo à medida que evolui a execução do programa é de fundamental importância. É necessário um desenvolvimento nesse sentido, com relação tanto a estruturas, desde as variáveis simples (atômicas) até as mais complexas, quanto a algoritmos ligados às estruturas, também dos mais simples até os de grau considerável de complexidade.

Assim, ferramentas capazes de exibir esquemas gráficos de abstração dessas estruturas, e das transformações nos dados associadas com a dinâmica de execução dos programas, podem exercer um papel decisivo no ensino de programação. Essa percepção tem motivado vários estudos e construção de ferramentas na área de Animação de Algoritmos e Estrutura de Dados, sempre se buscando tornar o aprendizado das disciplinas acadêmicas relacionadas mais estimulante e eficaz.

O projeto Balsa, Brown ALgorithm Simulator and Animator (Brown, 1988 apud. Garcia, 1997) e o projeto ZEUS (Brown, 1992) foram precursores desses estudos. A partir de então, vários outros projetos foram desenvolvidos como XTANGO (Stasko, 1992) e posteriormente o projeto POLKA. Além desses, destaca-se o software AIA (Algorithms in Action), que utiliza visualização para demonstração da execução dos algoritmos, correlacionando essa execução com os comandos correspondentes no código fonte (Stern, Naish e Sondergaard, 2000). Outros exemplos são o projeto ASTRAL (Garcia, Rezende e Calheiros, 1997), ODIN, ambiente de apoio ao ensino de lista encadeada (Madeira, 2005) e SPYKE, tutorial voltado ao ensino de pilhas e filas (Baldessar, 2003).

Embora uma análise comparativa dessas ferramentas extrapole o escopo deste trabalho, pode-se dizer que as ferramentas às quais se pode ter livre acesso via internet possuem, cada uma delas, pontos positivos e negativos. A maioria provê apenas a visualização das alterações sofridas na estrutura de dados (e/ou no seu conteúdo), para cada operação específica, sem, contudo, exibir simultaneamente o código correspondente à operação, e sua dinâmica de sua evolução, passo a passo. Algumas outras aliam as duas funcionalidades acima, mostrando o processo de “rastreamento” do código e as transformações efetivas nos dados (Algorithms in Action, por exemplo); entretanto, não existe uma funcionalidade que permita inserir comentários considerados necessários pelo professor em determinadas situações detectadas como pontos especiais de dificuldade³. Também sentiu-se falta nas ferramentas disponíveis de uma funcionalidade de grande utilidade para ensino de estruturas de dados, que é uma estratégia para simular o rastreamento de módulos recursivos, com exibição simultânea do código, e acompanhamento da execução linha a linha.

A partir dessa análise das ferramentas existentes, optou-se pelo desenvolvimento de protótipos de animações, através de editores de apresentações, onde se tem a possibilidade de combinar os recursos acima descritos (rastreamento e exibição das

³ A ferramenta *Algorithms in Action* possui uma janela que explicita o significado de cada linha selecionada na janela do código fonte, o que é diferente de fornecer explicações em diferentes níveis de abstração, conforme o caso.

estruturas) e ainda inserir comentários elucidativos com total flexibilidade e ainda exibir a saída esperada para a execução do algoritmo, quando for o caso⁴.

As Figuras 1 e 2 ilustram momentos de uso de duas das animações desenvolvidas. No primeiro caso, trata-se do rastreamento do algoritmo de percurso em árvore binária, método “pré-ordem” (visitar raiz, percorrer subárvore da esquerda, percorrer subárvore da direita). Nesse estágio de andamento do rastreamento, três nós já foram “visitados” (no caso, trata-se tão-somente de exibir o conteúdo) e o processo de empilhamento dos ponteiros de retorno é mostrado à esquerda da árvore (pilha **P**). Há um destaque em cores para a seta que designa o comando a ser executado no próximo passo (à esquerda do código) e para o ponteiro de trabalho (**q**), que no momento registrado aponta para o nó que contém o valor “D”.

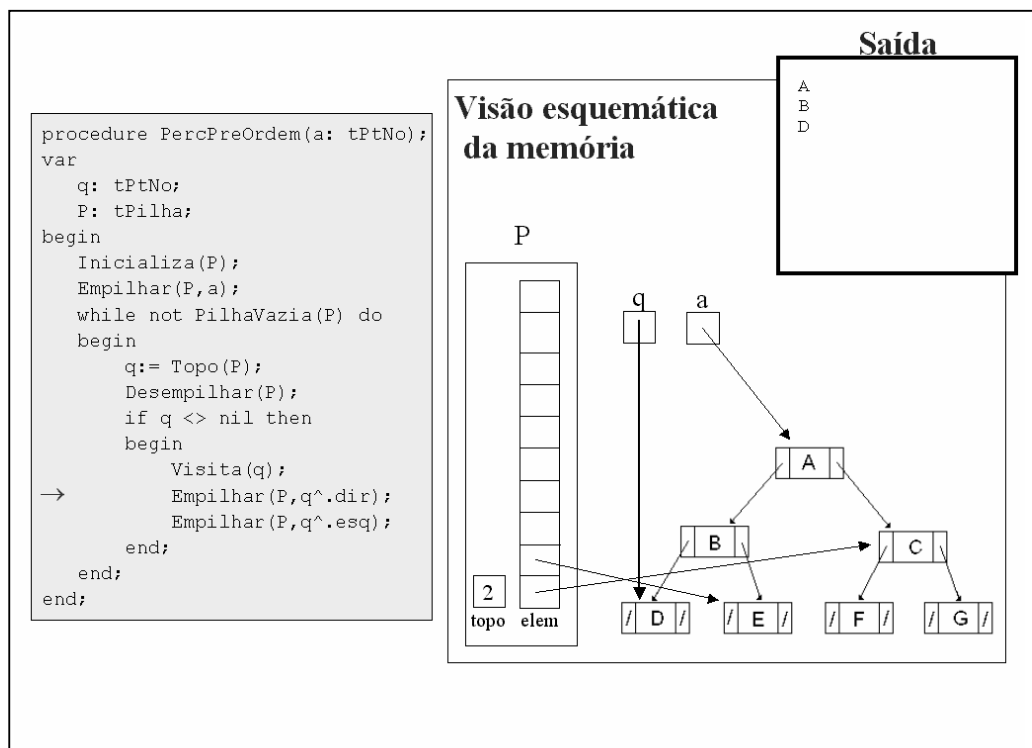


Figura 1. Rastreamento do algoritmo em pré-ordem de percurso em árvore binária.

Já a Figura 2 ilustra o rastreamento da versão recursiva do algoritmo empregado na animação anterior. Adotou-se a estratégia, considerada a priori mais intuitiva, de

⁴ O desenvolvimento e uso desses protótipos está servindo atualmente de subsídio a um projeto mais amplo de desenvolvimento, via programação, de uma ferramenta que reúna um amplo leque de recursos para o ensino de estruturas de dados. Uma das limitações da abordagem com editor de apresentação é não permitir que o usuário/aluno possa fornecer dados que serão utilizados, por exemplo em uma operação de inserção. Entretanto essa limitação não traz grande prejuízo, já que o valor a ser armazenado não é o ponto de relevância primordial, e sim o entendimento do processo para essa inserção.

uma sucessão de “folhas” para ilustrar cada ativação do módulo e da exibição da pilha subjacente à execução, com os sucessivos valores assumidos pelo parâmetro que designa a (sub)árvore cujo percurso está em processo. Cumpre ressaltar que nas *folhas* sobrepostas permanece a seta marcando o comando que ocasionou nova ativação e a partir do qual, continuará o processamento após o retorno (a execução da ativação *i* só prosseguirá quando for encerrada a ativação *i+1* e, por extensão a de todas as que lhe sucederem).

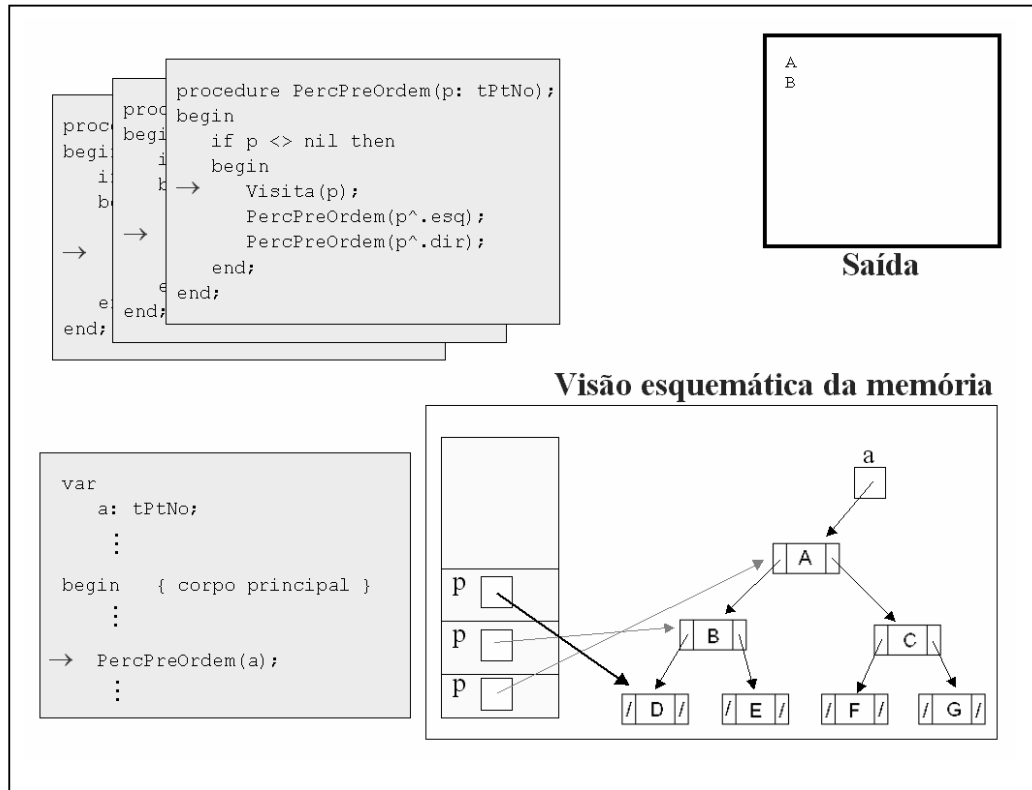


Figura 2. Rastreamento do algoritmo recursivo em pré-ordem de percurso em árvore binária.

4. Descrição da Experiência em Busca de uma Prática Inclusiva

De acordo com os princípios já explicitados, a busca de uma ação inclusiva em educação envolve a adoção de práticas concretas voltadas à correção de problemas detectados. Por outro, não pode prescindir do uso de tecnologias educacionais adequadas à situação, entendendo-se que uma sem a outra se esvazia. Esses elementos da experiência realizada são descritos nesta seção.

4.1. A Detecção do Problema

Várias referências foram utilizadas para identificação dos problemas no processo de ensino/aprendizagem. O primeiro exame individual foi um deles; antes disso, contudo, através das interações cotidianas com os alunos (esclarecimentos de dúvidas, supervisão

de exercícios propostos para sala de aula, e da própria explicitação das dificuldades enfrentadas), já vinha se dando o processo de diagnóstico, no qual foi identificado o seguinte: em primeiro lugar, alguns alunos apresentavam significativas dificuldades em conteúdos ligados à disciplina de programação precedente (Programação), notadamente no que diz respeito ao uso adequado de estruturas de controle em diversas situações, desde as mais simples até outras mais sutis, conforme o caso particular⁵. Em segundo lugar, observou-se uma dificuldade mais generalizada com relação à compreensão dos conceitos ligados às estruturas dinâmicas: uso adequado de ponteiros, alocação e liberação de memória, e o emprego efetivo desses conceitos para implementação de estruturas como listas encadeadas, os casos particulares de pilhas e filas, etc. Um outro tópico bastante associado a dificuldades de apreensão foi o uso da técnica de recursividade.

4.2. Ações Desencadeadas

Levando-se em conta o processo diagnóstico, foram organizadas as seguintes atividades e adotados os seguintes procedimentos: a) foi instituída uma aula extra semanal (duas horas) que serviria como um momento especial de retomada de todos os assuntos que estavam suscitando dúvidas, incluindo-se não só aqueles próprios da disciplina Estrutura de Dados I, mas também da disciplina que a antecede; b) para orientar o estudo dos alunos, conforme o grau de dificuldade de cada um, foram disponibilizadas listas de exercícios, buscando-se atender ao princípio da complexidade progressiva entre as listas e entre os exercícios de cada uma delas; c) associaram-se ao trabalho cinco monitores(as), por meio de um programa específico do curso⁶, cada um dos quais ficou responsável por assistir diretamente um conjunto definido de dois ou três alunos (um grupo de 12 alunos, de um total de 32 na turma, participou desse trabalho extra); d) Enfatizou-se o uso de animações gráficas, especialmente na apresentação de estruturas encadeadas por ponteiros e algoritmos recursivos. As ferramentas foram disponibilizadas via internet aos alunos e seu uso nos estudos subsequentes às aulas foi fortemente estimulado.

4.3. Resultados Obtidos

Os resultados podem ser expressos de forma qualitativa e quantitativa. A avaliação qualitativa dos resultados decorre do retorno (feedback) dado pelos alunos quanto à aplicação dos procedimentos antes descritos, de modo especial com relação à utilização das animações. Nesse sentido houve uma manifestação generalizada que o momento específico para tratar das dificuldades (a aula extra) foi fundamental, uma vez que alguns alunos, segundo seus depoimentos, sentem dificuldade em expressar suas dúvidas em turmas maiores, mas o fazem mais facilmente em um grupo mais reduzido. O uso de animações teve forte receptividade. Houve manifestações acerca do diferencial

⁵ Foi dedicada atenção especial a esses tópicos no diagnóstico, uma vez que os mesmos constituem pré-requisitos indispensáveis para a disciplina de Estruturas de Dados.

⁶ O programa ACC (Atividade Curricular Complementar) instituído formalmente no Curso Bacharelado em Ciência da Computação, prevê um conjunto bastante variado de atividades extracurriculares associadas à concessão de créditos. Dentre essas atividades, escolhidas conforme a preferência individual, está a atividade de monitoria.

positivo que a aplicação desse recurso proporcionou em termos de aprendizagem, seja no entendimento do uso de ponteiros, seja no entendimento da técnica de recursividade.

Já quanto aos resultados quantitativos, podemos destacar: a) o diagnóstico realizado até o primeiro exame individual, apontava que 20 alunos, de um total de 32 (62,5%) encontravam muitas dificuldades na aprendizagem; Esse número, após o trabalho empreendido, baixa para 28,13%, considerando-se o resultado geral da avaliação, onde apenas 9 não conseguiram alcançar aprovação na disciplina; b) Dentre os aprovados, 40,63% obtiveram conceito “bom” ou “excelente”, e 31,25% obtiveram conceito “regular”.

5. Considerações Finais

Este artigo descreveu uma experiência com a disciplina Estrutura de Dados, objetivando-se uma ação inclusiva. Os resultados obtidos podem ser considerados modestos, se entendermos que uma experiência bem sucedida de ação inclusiva deveria alcançar um percentual muito mais próximo de 100% para aprovações com excelente rendimento. De fato, essa é a meta de uma pedagogia verdadeiramente acolhedora.

Por outro lado, se considerarmos que parte dos problemas vivenciados em uma disciplina da natureza daquela com a qual se trabalhou está ligada a lacunas de aprendizagem relacionadas a disciplinas anteriores, a reflexão que se apresenta é a necessidade de se pensar o aprendizado de certas competências ou certos conhecimentos no âmbito de um curso de Computação como algo que extrapola os limites de uma disciplina. Portanto, uma continuidade de experimentos nesta linha deve considerar blocos de disciplinas, aplicando os princípios acima descritos de uma pedagogia inclusiva desde as iniciais, e buscando-se meios institucionais para que tais princípios sejam implementados, para além de iniciativas isoladas.

Dentro dessa perspectiva, e com as devidas correções derivadas de novas experiências, entende-se que a abordagem aqui seguida, onde o processo de avaliação deve não certificar, ou “separar o joio do trigo”, mas subsidiar o pleno desenvolvimento do educando, é bastante promissora.

Referências

- Baldessar, L. (2003) “SPYKE: Tutorial voltado ao ensino de pilhas e filas”. Universidade do Extremo Sul Catarinense – UNESC.
- Brown Marc H. (1988), “Exploring algorithms using BALSA-II”. IEEE Computer, May.
- Brown, Marc H. (1992) “Zeus: A System for Algorithm Animation and Multi-View Editing”. Systems Research Center, February.
- Byckling, Pauli; Sajaniemi, Jorma (2006) “Roles of Variables and Programming Skills Improvement”. In: Technical Symposium on Computer Science Education - SIGCSE 2006. Houston, Texas USA, March 1-5.
- Ferreira, Benedito de J. Pinheiro, "Ensino Inicial de Programação: Contribuições para uma Metodologia". In: XX SEPAl/III CTIC, Belém, PA, 24-27, outubro, 2006. Anais. pp. 82-90.

- Garcia I. C., Rezende, P. J. de, e Calheiros, F. C. (1997) “Astral: Um Ambiente para Ensino de Estruturas de Dados através de Animações de Algoritmos”. Revista Brasileira de Informática na Educação, Florianópolis, SC, v. 1, pp. 71-80.
- Kelleher, Caitlin; Pausch, Randy. *Lowering the Barriers to Programming: A Taxonomy of Programming Environments and Languages for Novice Programmers*. In : ACM Computing Surveys, Vol. 37, No. 2, June 2005, pp. 83–137.
- Luckesi, Cipriano Carlos (2005) “Avaliação da Aprendizagem na Escola: reelaborando Conceitos e Recriando a Prática”, 2^a. ed. rev. Salvador: Malabares Comunicação e Eventos. 115p.
- Madeira, M. (2005) “ODIN - Ambiente Web de Apoio ao Ensino de Estruturas de Dados Lista Encadeada”. Universidade do Extremo Sul Catarinense – UNESC.
- Olsen, Anne L (2005) “Using Pseudocode to Teach Problem Solving”. In: Journal of Computing Sciences in Colleges. Volume 21, Issue 2, Consortium for Computing Sciences in Colleges. December.
- Stasko, John T. (1992) "Animating Algorithms with XTANGO", SIGACT News, Vol. 23, No. 2, pp. 67-71.
- Stern L., and Naish, L. (2002) “Animating Recursive Algorithms Interactive Multimedia”, Electronic Journal of Computer Enhanced Learning, Volume 4, Number 2.