

ODIN - Ambiente Web de Apoio ao Ensino de Estruturas de Dados Lista Encadeada

Maicon Francisco Madeira¹, Priscyla Waleska Tagino de Azevedo Simões^{1,2}, Paulo João Martins

¹Departamento de Ciência da Computação – Universidade do Extremo Sul Catarinense (UNESC) - Av. Universitária, 1105 - Bairro Universitário - C.P. 3167 - CEP: 88806-000 – Criciúma – SC – Brasil

²Departamento de Medicina - Universidade do Extremo Sul Catarinense (UNESC) - Av. Universitária, 1105 - Bairro Universitário - C.P. 3167 - CEP: 88806-000 – Criciúma – SC – Brasil

`mmadera@pop.com.br, {pri,pjm}@unesc.net`

Abstract. *The present papers develops the ODIN – a supporting environment for linked list teaching, which aims to help its teaching-learning process, in the discipline of data structure, for computer science students. Its purpose is to graphically show how the primitive operations of linked list in the language C++ work, through an environment leading to the concept of computer science in education and following many ergonomic standards. With the objective of facilitating the abstraction's process by the students, besides providing the basic recourses for the teacher, allowing him teaching practice in the computer, where the students can to create and test their solutions out of class.*

Resumo. *O presente artigo refere-se ao ODIN – ambiente web de apoio ao ensino de lista encadeada, que visa auxiliar seu processo de ensino-aprendizagem na disciplina de estrutura de dados. Seu objetivo é demonstrar graficamente o funcionamento das operações primitivas de lista encadeada na linguagem C++, por meio de um recurso voltado aos conceitos de informática na educação, mais especificamente alguns critérios ergonômicos. Afim de facilitar o processo de abstração por parte do acadêmico, além de disponibilizar um recurso de apoio ao professor, permitindo ao mesmo efetuar a prática de ensino no computador, onde os alunos poderão desenvolver e testar suas soluções fora do horário de aula.*

1. Introdução

As estruturas de dados são utilizadas, hoje em dia, nas mais diversas áreas da computação, e seu entendimento é fundamental para qualquer acadêmico desta área. No entanto, a complexidade de certas estruturas torna difícil a tarefa de entendê-las por meio dos recursos habituais de ensino utilizados. Nestes casos, recursos visuais como uma animação, por exemplo, praticamente eliminam as dificuldades e tornam esta tarefa mais agradável.

Atualmente existe uma carência de recursos voltados para o apoio do processo de ensino-aprendizagem de estrutura de dados, principalmente com embasamento em

modalidades de aplicação em informática da educação, que utilizem métodos informatizados para a aquisição e aplicação do conhecimento, observando-se que os recursos existentes no meio computacional, em sua maioria, não foram projetados para esse fim.

2. Estruturas de Dados

Uma estrutura de dados é baseada na organização, representação e manipulação das informações contidas geralmente na memória principal / secundária do computador. Essas informações podem ser classificadas em instruções que comandam o funcionamento da máquina e determinam a maneira como os dados devem ser tratados, e estes dados, correspondem à porção das informações a serem processadas pelo computador [Moraes 2001].

O estudo das estruturas de dados compreende dois objetivos complementares, o teórico, que identifica e desenvolve modelos matemáticos, determinando que classes de problemas podem ser resolvidas, considerando então um Tipo Abstrato de Dados como um recurso a ser empregado durante a resolução de problemas em geral, e o prático, que cria uma representação adequada dos objetos para o desenvolvimento de rotinas capazes de atuar sobre essas representações, de acordo com o modelo avaliado, considerando a implementação por meio da abstração de dados, como um problema em si, que pode ser resolvido através de outros tipos de dados já disponíveis [Pereira 1996].

Pode-se representar uma ED por meio de pilhas, filas, árvores, listas encadeadas, entre outros tipos, afim de se obter uma resolução para várias classes de problemas devidamente abstraídos do mundo real, como também maior performance na execução de seus algoritmos.

2.1. Listas Encadeadas

As listas encadeadas são uma das estruturas de dados mais utilizadas no desenvolvimento de *programas* [Pereira 1996]. São recursos que utilizam mecanismos de acesso encadeado, que armazenam seus dados somente quando necessário, fazendo a liberação da memória quando não mais utilizada.

A popularidade das listas deve-se ao fato que, no decorrer de nosso dia-a-dia, é comum sua utilização para organização de tarefas, como por exemplo: lista de convidados para casamento, lista de aviões que devem decolar no aeroporto, a lista de processos de estado pronto de um certo sistema operacional, esperando pelo uso do processador, ou seja, a estrutura da lógica de pensamento já contempla a utilização desse tipo de organização no dia-a-dia [Pereira 1996].

Assim como em qualquer outro algoritmo, as funções primitivas de lista encadeada podem ser quantificadas em relação ao tempo de execução, o que é relevante, pois a velocidade de execução do programa é também um critério importante, destacando-se desta forma aplicações científicas onde processamentos pesados podem ocorrer [Preiss 2002].

2.2. A Eficiência Das Listas Encadeadas

Para que haja uma maior compreensão de como é possível analisar um algoritmo de listas encadeadas e determinar sua eficiência, primeiramente é necessário estudá-lo, e tirar conclusões sobre como sua implementação se comporta de modo geral [Lafore 1999].

As listas encadeadas têm como particularidade a inserção e remoção de elementos do início de sua cadeia, que utiliza um tempo $O^1(1)$ [Preiss 2002].

Para se fazer uma busca por elementos específicos em uma lista encadeada, é necessário analisar metade dos itens da lista, em média. Isto requer $O(N)$ comparações [Preiss 2002].

Para que se compreenda com clareza as estruturas de dados, é necessária uma boa experiência em uma linguagem de programação, pois, apresenta uma organização diferenciada de dados na memória, dificuldades no estudo e entendimento da sua lógica de funcionamento, e no que se referem suas características exclusivas para cada tipo de organização. Sendo assim, é necessário ter uma maior atenção no processo de ensino de estruturas de dados.

3. Ensino das estruturas de dados

O ensino de estruturas de dados não consiste em simplesmente mostrar para o acadêmico a ação das operações sobre as informações que a estrutura contém. É tentar apoiar o acadêmico a desenvolver sua lógica de construção de programas, em problemas que usam as estruturas de dados como forma de armazenamento das informações, ou como estruturas auxiliares na solução de problemas. Isto significa trabalhar com as estruturas de dados, onde as operações são apresentadas aos estudantes através das modificações que elas provocam quando são executadas, independentemente do meio que são armazenadas [Moraes 2001].

Na busca pela facilitação do processo de ensino - aprendizagem das estruturas de dados, podem ser encontradas algumas ferramentas para a demonstração da lógica de funcionamento das estruturas, como por exemplo, o EDDL que é um programa educativo no âmbito do ensino-aprendizagem das estruturas de dados [Azul and Mendes, 1997], o ASTRAL ambiente para o ensino de estrutura de dados através de animações de algoritmos [Calheiros et al, 1996] e o SPYKE tutorial voltado ao ensino de pilhas e filas, ambos concebidos seguindo alguns conceitos de informática na educação [Baldessar, 2003].

No que se refere ao processo de aprendizagem, pode ocorrer de duas formas: a informação simplesmente é memorizada, ou ela é trabalhada por meio de esquemas mentais e agregada a esses esquemas. Neste último caso, o conhecimento é construído. Essas diferenças em aprender podem ser fundamentais, pois por um lado, quando a informação não é trabalhada, e dificilmente poderá ser aplicada na resolução de problemas, podendo ser repetida de maneira mais ou menos fiel, indicando fidelidade da retenção. Por outro lado a informação trabalhada está armazenada nos esquemas mentais e será colocada em prática diante de situações problema ou desafios. Desta forma o aprendiz solucionará os problemas de modo a acrescentar e agregar mais conhecimento ao já existente [Valente 1993].

¹ Notação comumente utilizada para determinar a eficiência de um algoritmo por meio da função que expressa a relação entre n (tamanho) e t (tempo de execução), a fim de estimar a taxa de crescimento da função.

Conforme Cybis (2002), os ambientes educacionais se desenvolvidos utilizando-se de recomendações e recursos ergonômicos, podem trazer ganhos efetivos aos processos pedagógicos, pois esta integração entre propriedades de usabilidade e aprendizagem pode garantir maior sucesso no processo de aprendizagem.

4. Ergonomia de *Software*

O conceito de ergonomia começou a ser discutido com o surgimento da barreira da informática nos anos 80, isso porque, no início dessa década, os programas eram utilizados somente por seus desenvolvedores. Logo, estes programas começaram a atender um pequeno público externo que recebia um treinamento pesado para a utilização dos mesmos. O problema surgiu quando os programas começaram a ser utilizados pelo público mais amplo e sem treinamento, o que ocasionou uma quebra no modelo interface homem-computador, pois a falta de interesse pela lógica de utilização dos programas fazia com que as interfaces com os usuários fossem sempre o último item a ser observado no desenvolvimento dos programas [Cybis 2002].

O desenvolvimento de sistemas com boa usabilidade podem apoiar os usuários no desenvolvimento de suas tarefas, em se tratando da eficiência, eficácia e produtividade da interação. Com este conceito, o usuário poderá atingir seus objetivos com menos esforço e mais satisfação [Cybis 2002].

Com base em algumas recomendações ergonômicas apoiadas a preocupação de criar um recurso didático, para que possa apoiar o ensino-aprendizagem da estrutura de dados tipo lista encadeada, desenvolveu-se o ODIN portal web para apoio ao ensino de estrutura de dados lista encadeada, trabalho de conclusão de curso desenvolvido no curso de Ciência da Computação, da Universidade do Extremo Sul Catarinense, UNESC.

5. Sistema Desenvolvido

O protótipo de apoio ao ensino ODIN consiste na visualização das funções primitivas de listas encadeadas, demonstrando graficamente a execução dos algoritmos passo a passo, por meio de animações interativas sobre uma interface, onde os acadêmicos possam interagir e agregar conhecimento, almejando com isso, apoiar o processo de ensino - aprendizagem da temática em questão. A base metodológica utilizada para o seu desenvolvimento foi composta pelas seguintes etapas, que envolvem o levantamento bibliográfico anteriormente descrito, a elaboração de um questionário, desenvolvimento da lista encadeada, projeto de interface e fluxo de informações do programa, que serão descritas a seguir.

5.1. Elaboração de um Questionário

No início do desenvolvimento do TCC, foi elaborado um questionário, com a finalidade de quantificar as dificuldades que os acadêmicos encontram no entendimento das estruturas de dados, e também de tópicos correlacionados ao processo de ensino-aprendizagem nessa área de conhecimento.

Com relação aos resultados obtidos, observa-se que em todos os tópicos relacionados, não foram diagnosticadas deficiências graves em relação ao entendimento dos acadêmicos nos conteúdos abordados na disciplina de estrutura de dados do curso

de Ciência da Computação da UNESC. O resultado do questionário foi um ponto determinante na escolha da estrutura a ser desenvolvida no ODIN.

A lista encadeada foi escolhida em primeiro lugar por que ela pode utilizar técnicas de encadeamento, e, embora seja uma estrutura de ensino inicial na disciplina de ED, podemos implementá-la de diferentes modos, utilizando por exemplo, ponteiros, que de acordo com o gráfico nº 02 da figura 1, onde demonstra que 50% dos acadêmicos pesquisados encontram dificuldades em aplicar esse conceito. Em segundo lugar, também verificou-se que, apesar 65,63% dos acadêmicos pesquisados entenderem com clareza as operações primitivas de listas encadeadas, não compreendem com clareza a lógica sobre ponteiros, o que acaba gerando maiores dificuldades nas demais estruturas que utilizam.

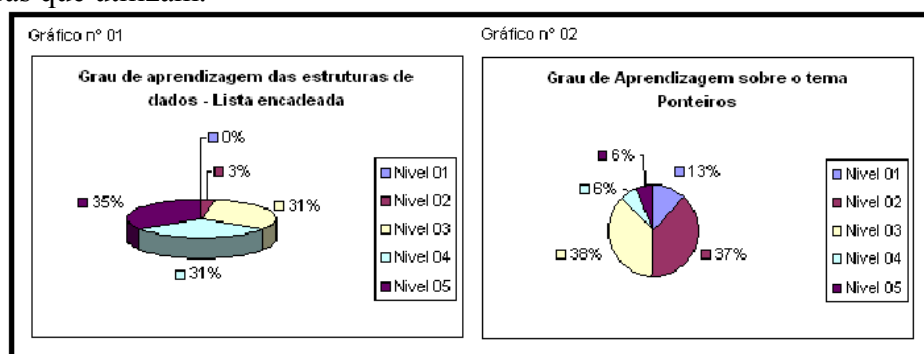


Figura 1. Gráficos do Questionário – Lista Encadeada e Ponteiros.

Esse questionário constitui-se de 08 questões, sendo elas, objetivas e de múltiplas escolhas, (Escala de Gauss), e foi aplicado em acadêmicos que já cursaram a disciplina de ED no curso de Ciência da Computação da UNESC.

5.2. Desenvolvimento da lista encadeada

No processo de ensino-aprendizagem de lista encadeada os acadêmicos apresentam dificuldades em entender como funciona a evolução da lógica das funções primitivas, por isso, foi projetado o ODIN que oferece aos usuários a demonstração de como são realizadas as operações primitivas básicas de uma lista, detalhando na tela o processo que não é visto ao se inserir ou remover elementos em um programa de lista encadeada.

Ainda nessa etapa, realizou-se um estudo das linguagens Java e C++ utilizadas para implementação das funções primitivas. A lista foi implementada em C++, pois é a linguagem padrão utilizada no curso de Ciência da Computação da UNESC. Na concepção do ambiente, ou melhor, das operações gráficas da lista, foi utilizada a Linguagem Java, em razão de sua operacionabilidade e fácil utilização em *browsers de internet* [Lafore, 1999].

5.3. Projeto de Interface

A interface do protótipo ODIN é composta por dois elementos principais, que são o espaço virtual (*Site*), voltado para o uso na *Internet*, e dentro dele, o aplicativo de simulação do funcionamento da lista encadeada. Ambos foram construídos obedecendo alguns dos conceitos de ergonomia de *software* e informática na educação.

Por ser um recurso de apoio ao ensino, o propósito da interface do programa é envolver o acadêmico, prender sua atenção aos conceitos abordados, de modo que o

mesmo se sintam bem ao executar as operações dentro do ambiente, por isso, o ODIN possui uma interface limpa, com uma linguagem clara e explicativa, focalizada nas tarefas em que o acadêmico irá desempenhar dentro do ambiente. Analisando-se a interface (figura 2), ressalta-se que não foram utilizadas cores fortes ou fontes pequenas, o que dificultaria a utilização por parte do acadêmico.



Figura 2. Tela Principal do ambiente ODIN.

5.4. Fluxo De Informações Do Sistema

O módulo do ODIN responsável pela parte de demonstração gráfica das funções primitivas de lista encadeada é composto de três partes, a da esquerda que demonstra o código fonte de cada função da lista, à direita onde serão desenhados os nós inseridos na estrutura de dados e a parte inferior que compreende os menus compostos de vários botões, que permitiram ao usuário a interação com o ambiente, conforme a figura 3.

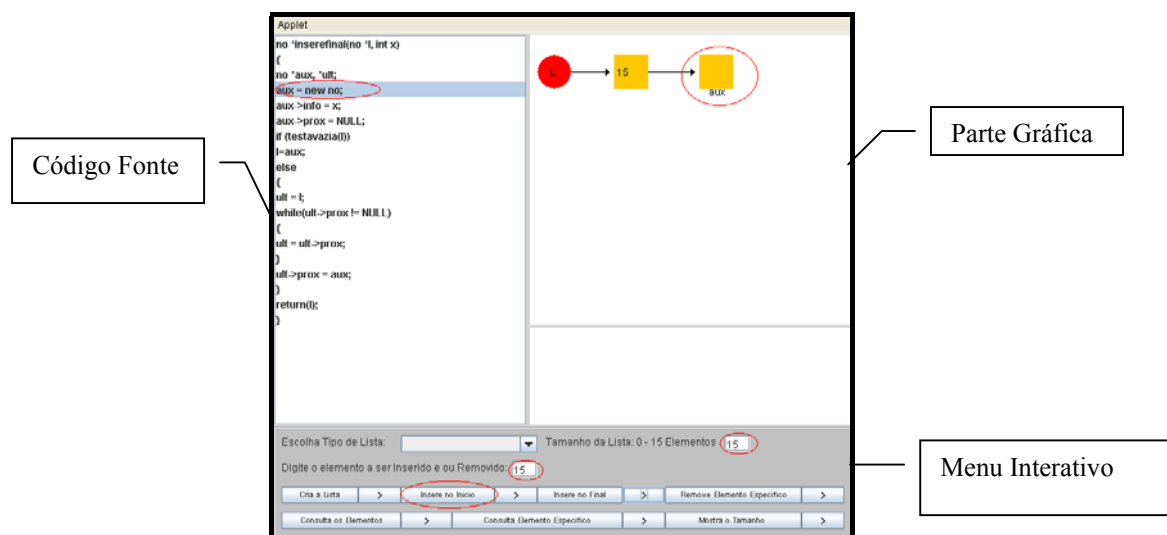


Figura 3. Tela do aplicativo ODIN.

Na tela principal do aplicativo, o usuário encontrará várias opções de interação como a lista encadeada. Primeiramente, o usuário irá definir com quantos elementos a estrutura irá trabalhar (de 0 à 15 elementos), e feito isso, poderá iniciar suas interações com o aplicativo, clicando sobre os botões que irão efetuar operações do tipo como por exemplo de criação da lista, inserção de elementos no inicio da lista, inserção de

educacionais, de forma a despertar a atenção dos acadêmicos e oferecer alternativas de pesquisa e processamento de informação.

Baseado neste conceito foi desenvolvido o ODIN, ambiente *web* de apoio ao processo de ensino - aprendizagem das listas encadeadas. Em sua concepção, houve uma busca pela utilização de abordagens sobre informática na educação, sendo utilizados alguns critérios ergonômicos, possibilitando-se a demonstração gráfica das funções primitivas dos algoritmos de lista encadeada.

Pretendeu-se com esse trabalho fomentar a idéia do desenvolvimento recursos voltados para o apoio ao ensino, motivando a criação de grupos de pesquisas, para efetuar estudos mais profundos em relação à abordagens de informática na educação, destacando a importância dos mesmos se atribuídos a construção de aplicativos educacionais, tendo em vista que houve uma certa dificuldade encontrada no desenvolvimento desse trabalho de relacionar os conceitos de informática na educação ao protótipo.

Recomenda-se para trabalhos futuros, acrescentar demonstrações de outros tipos de estrutura de dados como, por exemplo, árvores, *hashing*, além de oferecer recursos de como ocorre a alocação de memória no momento da execução dos algoritmos.

7. Referências

- AZUL, A. e MENDES, A. (1997) “EDDL: *software* educativo no âmbito do ensino-aprendizagem das estruturas de dados”. Universidade Católica Portuguesa e Universidade de Coimbra – Portugal.
- BALDESSAR, L. (2003) “SPYKE: Tutorial voltado ao ensino de pilhas e filas”. Universidade do Extremo Sul Catarinense – UNESC.
- CALHEIROS, F. et al. (1996) “ASTRAL: ambiente para o ensino de estrutura de dados através de animações de algoritmos”. Universidade Estadual de Campinas – UNICAMP.
- COX, K. (2003) “Informática na educação escolar”. Campinas, SP: Autores Associados.
- CYBIS, W. de A. et al. “Ergonomia em software educacional: a possível integração entre usabilidade e aprendizagem”. Disponível em: <http://www.unicamp.br/~ihc99/ihc99/AtasIHC99/art24.pdf>. Acessado em: 14 ago. 2004.
- GOODRICH, M. e TAMASSIA, R. (2002) “Estrutura de Dados e Algoritmos em Java”. 2. ed. Porto Alegre: Bookman.
- LAFORE, R. (1999) “Aprenda em 24 Horas estrutura de Dados e Algoritmos”. Rio de Janeiro: Campus.
- MORAES, C. (2001) “Estrutura de Dados e Algoritmos – Uma Abordagem Didática”. São Paulo: Berkeley.
- PEREIRA, S. (1996) “Estrutura de Dados Fundamentais: Conceitos e Aplicações”. São Paulo: Érica.
- PREISS, B. (2002) “Estrutura de Dados e Algoritmos: Padrões de Projetos Orientados a Objeto com Java”. Rio de Janeiro: Campus.
- TAJRA, S. (1998) “Informática na Educação: Professor na Atualidade”. São Paulo: Érica.
- VALENTE, J. (1993) “Liberando sua mente: computadores na Educação Especial”. Campinas, Unicamp.