

# Avaliação de Interfaces de Ferramentas Computacionais para o Ensino de Estruturas de Dados e Algoritmos em Grafos: Heurísticas de Usabilidade\*

Rodrigo Pereira dos Santos<sup>1</sup>, Heitor Augustus Xavier Costa<sup>2</sup>, André Luiz Zambalde<sup>3</sup>

Departamento de Ciência da Computação – Universidade Federal de Lavras (UFLA)  
Caixa Postal 3037 – 37.200-000 – Lavras – MG – Brasil

<sup>1</sup>rpsantos@comp.ufla.br, <sup>2</sup>heitor@ufla.br, <sup>3</sup>zamba@ufla.br

**Abstract.** *Several computing tools are developed with the objective of decreasing difficulties of algorithms and programming learning. However, its use is not expressive, that is demonstrated by low assimilation subject indexes, which generate problems for advanced disciplines. So, this article evaluates interfaces of tools, through usability heuristics, looking for to verify the contribution of that aspect for this negative reality in Computer Science field.*

**Keywords:** *computing in education, human-computer interface, data structures and algorithms in graphs.*

**Resumo.** *Com o objetivo de facilitar a aprendizagem de disciplinas das áreas de algoritmos e programação, diversas ferramentas computacionais são desenvolvidas. Entretanto, seu uso não é expressivo, fundamentalmente pelas dificuldades inerentes as disciplinas e mesmo por problemas de interfaces. Dessa forma, este artigo visa analisar a interface de algumas ferramentas, através de heurísticas de usabilidade, buscando verificar a contribuição desse aspecto para esta realidade negativa na área de Computação e Informática.*

**Palavras-chaves:** *informática na educação, interface humano-computador, estruturas de dados e algoritmos em grafos.*

## 1. Introdução

Com o intuito de produzir melhores resultados no processo de aprendizagem em cursos da área de Computação e Informática, faz-se constante a necessidade de atualização das didáticas de ensino de forma geral, procurando transformar processos abstratos em concretos através do uso de *software* e possibilitando melhor compreensão dos tópicos abordados e maior interação entre aluno-objeto de trabalho. Uma das principais barreiras encontradas pelo aluno está nos processos iniciais dos cursos relacionados, no contato com a programação de estruturas de dados e de algoritmos em grafos.

Segundo [PIMENTEL, FRANÇA e OMAR 2003], diversos tipos de ferramentas têm sido propostos com o objetivo de facilitar o aprendizado de lógica e de programação, tais como mini-linguagens, mundos programáveis, ambientes de desenvolvimento controlados e ferramentas de animação. Porém, um problema enfrentado em muitas instituições é o baixo índice de assimilação dos estudantes nas disciplinas que exigem o conhecimento de programação, uma vez que a aprendizagem ocorre durante o curso.

---

\* Apoio PBIICT/FAPEMIG.

Segundo ([SANTOS e COSTA 2005]), as ferramentas computacionais mais comuns e procuradas pelos discentes estão relacionadas às animações do funcionamento de algoritmos, mas não basta apenas adotá-las e/ou sugerir-las durante as aulas. Deve ser realizada uma avaliação para verificar se tais ferramentas atendem às necessidades dos discentes, principalmente com relação à interface; pois os alunos enfrentam dificuldades com relação à interação e ao entendimento de ambientes educacionais. Dessa forma, várias métricas de interface devem ser seguidas durante o processo de desenvolvimento do *software* educacional para obter um produto usável e de sucesso para o que se propõe.

O objetivo do artigo é apresentar uma avaliação da interface de algumas ferramentas computacionais para o ensino de estruturas de dados e para o ensino de algoritmos em grafos, com base em determinadas heurísticas de usabilidade<sup>1</sup>, propostas por [NIELSEN 1993].

O artigo está organizado da seguinte forma: a seção 2 retrata a importância da interação humano-computador e o uso de técnicas de avaliação de interface como forma de manter a qualidade de *software* educacional; a seção 3 apresenta exemplos de avaliação de interfaces de ferramentas para o ensino de estruturas de dados e de algoritmos em grafos; e a seção 4 apresenta os procedimentos metodológicos realizados para o desenvolvimento deste trabalho. Por fim, são apresentadas as conclusões.

## 2. A Importância da Interação Humano-Computador e suas Técnicas de Avaliação

Segundo [ACM SIGCHI 1992], interação humano-computador é a disciplina preocupada com o *design*, avaliação e implementação de sistemas computacionais interativos para uso humano e com o estudo dos principais fenômenos ao redor deles. A interface é um elemento da interação humano-computador que envolve aspectos computacionais e como estes influenciam o usuário na realização de uma tarefa, utilizando o computador da maneira mais fácil e transparente possível.

Pelo exposto, pode-se definir a interface como um sistema de comunicação que possui dois componentes: i) um componente físico no qual o usuário percebe e manipula; e ii) um componente conceitual que interpreta, processa e raciocina ([ORTH 2005]).

De forma objetiva, [ZAMBALDE e ALVES 2004] afirmam que o cuidadoso estudo da comunidade de usuários e do conjunto de tarefas é a base para o estabelecimento dos objetivos da interface com o usuário. Os cinco fatores imprescindíveis a considerar são: i) **facilidade de aprendizado**: tempo e esforço para uma comunidade típica de usuários aprender o sistema; ii) **velocidade de resposta**: tempo de resposta a um determinado conjunto de tarefas; iii) **taxa de erros dos usuários**: quantos e quais os tipos de erros as pessoas cometem quando usam a interface; iv) **retenção com o tempo**: quanto tempo os usuários levam para reter o conhecimento sobre a interface; v) **satisfação subjetiva**: o quanto usuários gostam de utilizar o sistema. As respostas podem ser obtidas através de entrevistas ou questionários impressos que incluam escalas de satisfação.

O *design* de interface definido por [SHNEIDERMAN 1998] descreve um modelo metafórico baseado em três “pilares”. O *designer* deve utilizar um conjunto de *guidelines* que o oriente no desenvolvimento de interfaces (primeiro “pilar”). Seguindo esses padrões,

---

<sup>1</sup> Usabilidade é a propriedade de uma interface que permite classificá-la quanto à facilidade de aprendizado; eficiência de uso/desempenho na execução de tarefas; retenção com o tempo; minimização de erros; e satisfação subjetiva ([SHNEIDERMAN 1998]).

ele constrói protótipos de interfaces (segundo “pilar”). O terceiro “pilar” é dedicado aos testes de usabilidade – avaliação de especialistas e testes com usuários.

A avaliação de interface é um passo muito importante no processo de *design* de *software* interativo. Nesse sentido, segundo [ROCHA e BARANAUSKAS 2000], os testes de usabilidade têm o objetivo de medir quantitativamente o valor alcançado pelo sistema em cada um dos fatores de usabilidade de interesse. Tais testes são classificados em: i) métodos de inspeção; e ii) testes empíricos com usuários. Um dos métodos de inspeção é a avaliação heurística, ou seja, a avaliação de alguns princípios reconhecidos de usabilidade. [NIELSEN 1993] propõe um conjunto de heurísticas que dispostas em uma tabela permite a adequada inspeção de usabilidade de uma interface.

### 3. Procedimentos Metodológicos

Neste trabalho, foi elaborada pelos autores uma tabela, adaptada de [NIELSEN 1993] (Tabela 1) para análise de algumas ferramentas computacionais de ensino de algoritmos e de algoritmos em grafos. Para cada heurística, devem ser informados os problemas relacionados e a nota do avaliador entre 0 (RUIM) e 10 (ÓTIMO). Isso permite obter um *score* em porcentagem (10 heurísticas, cada uma com pontuação máxima 10) e, por conseguinte, a avaliação da interface, a qual será considerada satisfatória acima de 80%, razoável entre 50% e 79% e insatisfatória abaixo de 49%. A avaliação foi realizada por pesquisadores da área de Ciência da Computação entre março e setembro de 2006.

**Tabela 1 – Formulário de inspeção de usabilidade**

HEURÍSTICAS	PROBLEMAS DETECTADOS	MÉDIA DO AVALIADOR
(1) Visibilidade do <i>status</i> do sistema *o sistema informa o que está acontecendo?		
(2) Concordância entre o sistema e o mundo real *o sistema utiliza a linguagem do usuário?		
(3) Controle e liberdade ao usuário *o sistema apresenta facilidade de interação e “saídas” claras?		
(4) Consistência e padrões *diferentes situações ou ações representam a mesma coisa?		
(5) Prevenção de erros *o projeto prevê situação de erro ao invés de usar mensagem?		
(6) Reconhecer ao invés de lembrar *as telas usam metáforas (facilita memorização de instruções)?		
(7) Flexibilidade e eficiência de uso *o sistema atende a usuários experientes?		
(8) Projeto minimalista e estético *as informações são sintéticas e completas?		
(9) Reconhecimento, diagnóstico e recuperação de erros *problemas/soluções são facilmente indicados?		
(10) Ajuda e documentação *existem manuais simples e objetivos?		

### 4. Avaliação das Ferramentas Computacionais

Nesta seção, apresenta-se a avaliação da interface de algumas ferramentas computacionais para o ensino de programação, com base nas heurísticas da Tabela 1. A saber:

- **ASTRAL** – *Animation of Data Structures and Algorithms* ([GARCIA, REZENDE e CALHEIROS 1997]). Ambiente de programação para produção de animações de algoritmos, estruturas de dados e teoria dos grafos, desenvolvido no Instituto de Computação da UNICAMP. Na avaliação realizada, os seguintes problemas foram detectados: ausência de informações sobre os passos dos algoritmos em execução e os elementos visualizados; na ordenação, uso de pontos dispersos na tela durante a animação, ao invés de números; execução rápida e contínua, causando *stress* ao usuário

e não uso de metáforas para representar botões e outros recursos de interação; são apresentadas mensagens de erro, se não for selecionado o vértice inicial (em grafos); os problemas ocorridos durante a interação com o usuário são listados, mas sem exibir soluções; não existe documentação, exceto textos sobre algoritmos em grafos. Como pontos positivos: utiliza linguagem do usuário, oferece suporte a usuários experientes e mantém consistência a situações comuns, evitando o usuário adivinhar se diferentes palavras ou ações significam a mesma coisa;

- **TBC-AED** (Treinamento Baseado em Computador para Algoritmos e Estruturas de Dados) ([SANTOS e COSTA 2005]) e **TBC-GRAFOS** (Treinamento Baseado em Computador para Algoritmos em Grafos) ([SANTOS e COSTA 2006]). *Software* educacional que analisa tópicos básicos de programação, acompanhado de processo gráfico passo a passo, representando um repositório didático com visualização gráfica. Apresenta os seguintes problemas: ausência de informações sobre os passos dos algoritmos em execução; execução não pode ser interrompida durante o processo; as cores utilizadas durante o processo gráfico, em diferentes telas, representam situações diferentes; ausência de teclas de atalho; repetição das instruções de uso do programa. Como pontos positivos: o sistema utiliza a linguagem do usuário; apenas botões necessários em determinado momento são exibidos; *links* explicativos evitam a necessidade de aprendizagem via tutorial; processo gráfico com elementos numéricos melhora a visualização e o entendimento; legendas explicativas ilustram as etapas do processo de apresentação dos algoritmos;
- **TED** – Tutorial de Estruturas de Dados ([FLÁVIO 2004]). É um tutorial para dinamizar o ensino na disciplina de estrutura de dados. Apresenta os seguintes problemas: ausência de textos explicativos sobre o *status* do programa; ausência de metáforas para representar botões e outros recursos de interação; ausência de teclas de atalho; a execução é contínua, sem possibilitar paralisar o processo; e não existe documentação. Como pontos positivos: visualização dos acontecimentos durante a execução; uso da linguagem do usuário; e fácil interação;
- **DIDAGRAPH** ([DAGDILELIS e STRATZEMI 1998]). Permite o acompanhamento de animações da execução de algoritmos, em alto nível, por meio de uma visualização do grafo com o algoritmo em andamento e de uma descrição em linguagem de alto nível do algoritmo. Apresenta os seguintes problemas: ausência de textos explicativos sobre o *status* do programa; ausência de metáforas para representar botões e outros recursos de interação; a execução é contínua, sem possibilitar paralisar o processo; e não existe documentação. Como pontos positivos: visualização dos acontecimentos durante a execução; e uso da linguagem do usuário.

A Tabela 2 apresenta o resultado da avaliação dos pesquisadores.

**Tabela 2 – Resultados da avaliação quantitativa**

HEURÍSTICAS	MÉDIA DOS AVALIADORES			
	ASTRAL	TBC-AED e TBC-GRAFOS	TED	DIDAGRAPH
(1)	3	7	9	8
(2)	10	10	10	10
(3)	7	8	10	9
(4)	10	8	10	10
(5)	7	10	10	7
(6)	3	10	7	7
(7)	10	8	8	8
(8)	7	9	7	7
(9)	7	10	10	10
(10)	4	10	8	8
<b>TOTAL</b>	<b>68</b>	<b>90</b>	<b>89</b>	<b>84</b>

## 5. Conclusões

Conclui-se que dentre os produtos de *software* analisados, o TBC-AED, TBC-GRAFOS, TED e DIDAGRAPH alcançaram resultados satisfatórios (acima de 80%), uma vez que atendem aos critérios propostos por [NIELSEN 1993] e às pontuações propostas pelos pesquisadores. Entretanto, o ambiente ASTRAL foi classificado como razoável, conforme problemas apresentados na seção anterior.

Assim sendo, a preocupação com a interface deve ser uma constante durante o desenvolvimento de produtos de *software* educacionais, tanto quanto a intenção de melhoria do processo de aprendizagem. Essa meta conduz ao uso e aceitação de ferramentas desse tipo, ao produzir atratividade sobre aluno.

O presente trabalho consistiu em um estudo exploratório sobre usabilidade em ferramentas computacionais voltadas ao ensino de estruturas de dados e ao ensino de algoritmos em grafos. Como trabalhos futuros, serão avaliadas outras ferramentas abordadas em [SANTOS e COSTA 2005] e [SANTOS e COSTA 2006]. Além disso, serão realizados testes empíricos com usuários e propostas métricas para avaliação de interface de produtos de *software* educacionais.

## Referências

- ACM SIGCHI (1992). “Curricula for Human-Computer Interaction”. Technical Report. NY: ACM. Disponível em <http://www.acm.org/sigchi>.
- Dagdilelis, V. e Stratzemi, M. (1998) “DIDAGRAPH: Software for Teaching Graph Theory Algorithms”. ITiCSE 1998, Integrating Technology into Computer Science Education. Dublin, Ireland, 1998.
- Flávio, D. (2004) “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 jul 2006.
- Garcia, I. C., Rezende, P. J. 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 (RBIE'1997) nº 01, <http://www.inf.ufsc.br/sbc-ie/revista/nr1/garcia.htm>. Ambiente ASTRAL disponível em: <http://www.dcc.unicamp.br/~rezende/ASTRAL/>. Acesso: 28 abr 2005.
- Nielsen, J. (1993) “Usability Engineering”. Academic Press, Cambridge, MA.
- Orth, A. I. (2005) “Interface Homem-Máquina”. Ed. AIO – Porto Alegre – RS.
- Pimentel, E. P., França, V. F. e Omar, N. (2003) “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.
- Rocha, H. V. e Baranauskas, M. C. C. (2000) “Design e Avaliação de Interfaces Humano-Computador”, IME-USP, São Paulo, SP.
- Santos, R. P. e Costa, H. A. X. (2005) “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.
- Santos, R. P. e Costa, H. A. X. (2006) “Um Software Gráfico Educacional para o Ensino de Algoritmos em Grafos”. Conferência Ibero Americana WWW/Internet (CIAWI'2006). Murcia - Espanha.
- Shneiderman, B. (1998) “Designing the User Interface: Strategies for Effective Human-Computer Interaction”, 3ª ed., Addison-Wesley.
- Zambalde, A. L. e Alves, R. M. (2004) “Interface Homem-Máquina e Ergonomia”, Textos Acadêmicos para o Curso de Pós-Graduação ‘Lato Sensu’ (Especialização) à Distância – Administração de Sistemas de Informação. UFLA/FAEPE.