

# APRENDENDO LÓGICA DE PROGRAMAÇÃO VIA WEB

**Romero Tori**

Universidade de São Paulo–Escola Politécnica  
INTERLAB–Laboratório de Tecnologias Interativas-USP  
Instituto Sumaré de Educação Superior  
[rometori@usp.br](mailto:rometori@usp.br)

**Cilene Aparecida Mainente**

imes-Centro Universitário Municipal de São Caetano do Sul- São Paulo  
INTERLAB–Laboratório de Tecnologias Interativas-USP  
[cilenemainente@imes.com.br](mailto:cilenemainente@imes.com.br)

O ensino de lógica de programação é de extrema importância para todas as habilitações da engenharia, sendo essencial nos cursos de engenharia de computação. Apesar de possuírem conteúdo extremamente simples, as disciplinas que cobrem tal assunto, costumam ser enfrentadas com grandes dificuldades pelos aprendizes, e muitas vezes não são capazes de evitar graves deficiências normalmente apresentadas pelos alunos durante e após o curso. O problema nesse tipo de disciplina não está propriamente no conteúdo a ser passado mas nas habilidades a serem desenvolvidas. O raciocínio lógico, a visualização de algoritmos, entre outras, são habilidades que precisam do envolvimento do aluno em atividades práticas bem conduzidas, nas quais se possa aprender fazendo. Uma maneira de aprimorar o aprendizado de lógica de programação é através da utilização de recursos de visualização e animação de algoritmos. Este trabalho apresenta uma proposta de utilização de uma ferramenta para animação de algoritmos escritos em JAVA, em disciplinas de lógica de programação para engenharia. A ferramenta, desenvolvida na Universidade de Helsinki na Finlândia, se baseia no uso da WEB, permitindo o desenvolvimento de atividades práticas em laboratório, via intranet ou Internet.

## 1. Introdução

Em cursos de Ciência da Computação, o projeto e algoritmos é disciplina fundamental. Segundo [Mei97], as inovações na tecnologia de informação não são possíveis sem ancorá-las em algoritmos. Enquanto isto, muitos programadores ainda fazem de um editor de textos sua ferramenta de trabalho, visualizando a execução do código fonte através da chamada a procedimentos de impressão inseridos ao longo do código.

A busca pelo uso adequado da tecnologia para minimizar as deficiências e dificuldades encontradas no ensino de conceitos abstratos de programação, pretende colaborar para que o aprendiz desenvolva habilidades no manuseio de complexos algoritmos dentro do princípio de aprender fazendo. A natureza hipermídia da Internet permite criar materiais altamente interativos, com ligações a diferentes conteúdos espalhados pelo mundo. Além disto, tem-se facilidades de introdução de materiais hipermídia para ensino e de intercâmbio de informações entre aprendizes em locais distantes entre si.

“Ao aproximar o projeto e testes de algoritmos de uma experiência multimídia, através da utilização de sistemas de visualização e animação, estamos fornecendo condições para que o programador e o pesquisador de Ciência da Computação sintam-se livre para explorar tarefas mais interessantes e manusear problemas cada vez mais complexos “ [Sta98].

## 2. Ferramentas de visualização e animação

Segundo Brown[Bro94], a área de visualização de algoritmos possui estreita relação com as seguintes áreas da Ciência da Computação: animação, projeto de interface computação gráfica, tecnologia e engenharia de software.

Myers[Mye90] classifica os sistemas de visualização segundo o “aspecto” do programa que está sendo ilustrado (código, dados, algoritmo) e o tipo de imagem exibida (dinâmica, estática).

Os sistemas dinâmicos de visualização ilustram o estado atual das variáveis do programa através de figuras gráficas, geradas à medida em que o programa está sendo executado. Isto já não é possível nos sistemas estáticos.

Finalmente, os sistemas de visualização de algoritmos utilizam gráficos que mostram de forma abstrata como o programa opera. A diferença entre este tipo de visualização e a visualização de dados e códigos é que aqui pode não haver uma correspondência direta entre as figuras e os dados do programa, e as mudanças nas figuras podem não corresponder a partes específicas do código do programa.

Os sistemas de visualização dinâmica são referidos como sistemas de animação de algoritmos, por mostrarem a evolução na execução de um algoritmo. O sistema Jeliot [Haa97] é um exemplo de sistema de animação de algoritmo., fornecendo a visualização de código, das estruturas de dados e do algoritmo. Esse foi o sistema escolhido pelos autores para a pesquisa de uso da WEB no ensino/aprendizagem de lógica de programação.

O sistema Jeliot oferece um ambiente de visualização completo, no qual o usuário pode escrever seu programa Java que será interpretado e executado pelo Jeliot na Web. Além disso, se o programa utiliza os tipos de dados fornecidos pelo Jeliot, a animação será automaticamente gerada. Neste caso, o usuário não precisa desviar sua atenção para a inserção de anotações no algoritmo para a geração da animação.

## 3. O sistema Jeliot

O sistema Jeliot foi desenvolvido na Universidade de Helsinki na Finlândia e utiliza a linguagem Java na criação de programas a serem animados.

No Jeliot não é necessário escrever linhas adicionais de código para gerar a animação. Todas as rotinas necessárias para a criação da animação ficam encapsuladas nas operações dos tipos de dados fornecidos pelo Jeliot. Na versão atual é possível animar tipos primitivos, matrizes, pilhas e filas.

A apresentação do sistema é baseada na metáfora teatral, introduzida por Negroponte [Neg91], onde o programa a ser animado é o *script* da peça e os atores são as entidades gráficas que representam as variáveis do programa. A execução do programa pode ser vista em diversas janelas ou palcos simultaneamente. O diretor da peça ou o usuário, controla a aparência de cada entidade gráfica e configuração dos palcos.

A figura 1 ilustra um exemplo de animação no Jeliot, no momento da comparação entre o elemento de índice 3 do vetor(94) e a variável big(56). A palavra *yes* entre os valores indica que a condição é verdadeira. A linha de código que está sendo animada (`if table[i] > big`) permanece em destaque na área onde está o programa.

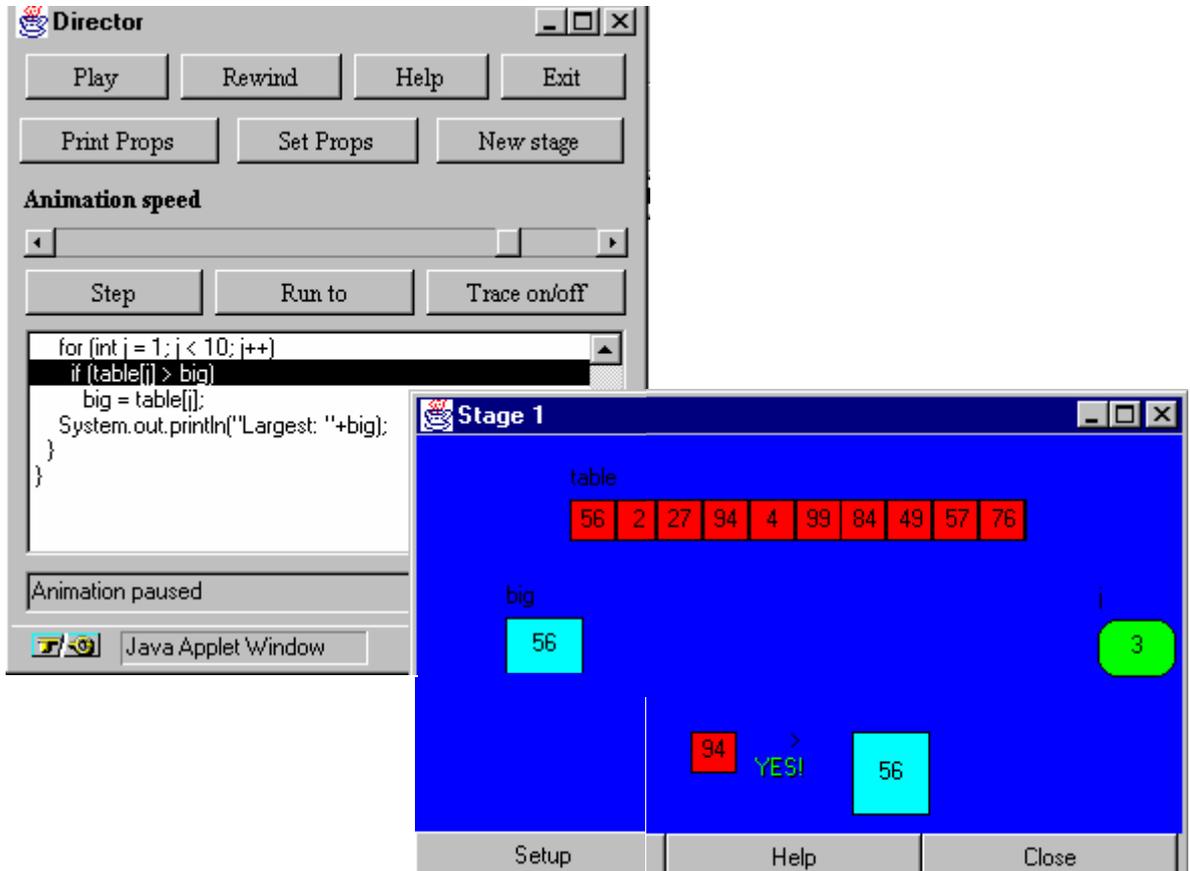


Figura 1 – Animando algoritmos com o Jeliot

A variável *i* está representada pela caixa arredondada no canto direito, a variável *big*, por um quadrado no lado esquerdo e o vetor *table*, por uma barra horizontal.

A figura 2 mostra a estrutura do modelo cliente servidor do Jeliot. Através da janela principal (1), o algoritmo é lido e enviado para a pré-compilação, consistindo de *parsing*(2) e geração de código(3). As fases (4)e(5) incluem rotinas de animação ao código fonte e extraem as variáveis que podem ser animadas. Essas variáveis são enviadas para a janela principal, podendo ter sua aparência configurada e ser incluídas em palcos.

O servidor passa o código gerado para o compilador Java(4) que gera um *applet* de animação que, juntamente com o Animador (6), mostra a animação.



- a) elaboração de uma nova proposta pedagógica para o ensino de lógica de programação, baseada no uso de animações de algoritmos;
- b) aprimoramento das ferramentas hoje disponíveis para animação de algoritmos.

## 5. Conclusão

Foi apresentado o ambiente de animação de algoritmos Jeliot desenvolvido na Universidade de Helsinki na Finlândia. Tal ambiente permite desenvolver e animar programas escritos em Java diretamente na Web. O uso de tal sistema por aprendizes de lógica de programação em complemento aos métodos de ensino tradicionais, pretende contribuir para o aprendizado dentro do princípio de aprender fazendo.

O próximo passo será efetuar estudos da influência da utilização do sistema Jeliot com alunos de cursos de Ciência da Computação e Engenharia de Computação e elaborar novas propostas pedagógicas para o ensino de lógica de programação, além de se propor e desenvolver aprimoramentos nas ferramentas para animação de algoritmos hoje existentes.

## Bibliografia

- [Bro94] BROWN, M.H.; STASKO, J.; DOMINGUES, J.; PRICE, B. (1994) Software visualization. Proceedings of the **CHI'94 Conference on Human Factors in Computing Systems**, 463
- [Haa97] HAAJANEN, J. *et al.* (1997) **IEEE**. Animation of user algorithms on the Web. Proc. VL'97, IEEE Symposium on Visual Languages, 356-363. EUA
- [Mei97] MEISALO, V. *et al.* (1997) Teaching algorithms with animation: a case study using Eliot. In Proc. **LeTTET'96, Learning Technology and Telematics in Education and Training**, 79-84. Joensuu Yliopistopaino
- [Mye90] MYERS, B.A. (1990) **Journal of Visual Languages and Computing**. Taxonomies of visual programming and program visualization(1), 97-123
- [Neg91] NEGROPONTE, N. (1991) **MIT**. Beyond the desktop metaphor. In Research Directions in Computer Science: An MIT Perspective (A. Myers *et al.*) MIT Press, Cambridge, MA, 183-190
- [Sta98] STASKO, J.T. *et al.* (1998) **MIT Press**. Software Visualization