

iVProg – Uma Ferramenta de Programação Visual para o Ensino de Algoritmos

Leônidas de Oliveira Brandão¹, Anarosa Alves Franco Brandão², Romenig da Silva Ribeiro¹

¹Instituto de Matemática e Estatística – Universidade de São Paulo
São Paulo – SP – Brasil

²Escola Politécnica – Universidade de São Paulo
São Paulo – SP – Brasil

{leo, romenig}@ime.usp.br, anarosa.brandao@poli.usp.br

Abstract. *The teaching of algorithms and techniques for problem solving are key elements in Introduction to Computing courses. However, students used to face several difficulties until learning and understanding the strategies to solve problems with your computer. The language syntax is one of the complicating factors in this process. Thus this paper presents a visual programming tool to reduce the cognitive overload related to syntax and brings, in addition to this, other benefits for students.*

Resumo. *O ensino de algoritmos e técnicas para resolução de problemas são elementos chave nos cursos de Introdução à Computação. No entanto existem muitas dificuldades que precisam ser transpostas pelos estudantes para que consigam aprender e compreender as estratégias para resolver problemas com o computador. A sintaxe da linguagem é um dos fatores complicadores desse processo. Dessa forma este trabalho apresenta uma ferramenta de programação visual para diminuir a carga cognitiva referente a sintaxe e que traz, além deste, outros benefícios para os alunos.*

1. A Ferramenta de Programação Visual iVProg

A ferramenta iVprog visa diminuir a sobrecarga cognitiva para os alunos das disciplinas de Introdução a Programação, a fim de permitir que eles se concentrem em organizar o pensamento e criar boas estratégias para resolver problemas. Pretende-se com isso aumentar o engajamento, motivação, aprendizado e aproveitamento dos alunos.

No geral, sabe-se que os professores dos cursos de introdução a computação gastam mais tempo ensinando detalhes de linguagens (estruturas e sintaxe) do que com o processo de programação propriamente dito [Caspersen 2009]. Tendo isso em mente, foi desenvolvida uma ferramenta baseada no paradigma de *Programação Visual*, o **iVProg**, com o intuito de diminuir a sobrecarga cognitiva referente ao aprendizado da sintaxe da linguagem e de ambiente de compilação. O iVProg permite que o aprendiz se concentre no problema proposto, focando-se na tentativa de projetar um algoritmo que o resolva. Consequentemente, o iVProg permite ao professor dar maior atenção ao ensino das estratégias para reconhecer e resolver problemas do que aos detalhes da linguagem.

O iVProg foi desenvolvido a partir do sistema *Alice* [Conway e Pausch 2009] durante o mestrado de Reginaldo Rideaki Kamiya sob orientação do primeiro autor, no

ano de 2009, no Instituto de Matemática e Estatística da Universidade de São Paulo [Kamiya 2009]. A motivação para utilizar o Alice como base se deu principalmente por ser uma ferramenta escrita na linguagem *Java* e de código livre. Um dos objetivos da ferramenta era, também, que funcionasse nos navegadores de Internet como um *applet* e fosse integrada a *Sistemas Gerenciadores de Curso (SGC)*, particularmente ao *Moodle*. Dessa forma, a linguagem *Java* satisfazia os requisitos e o código livre possibilitou que o Alice fosse utilizado como base inicial da primeira versão do iVProg. Hoje estamos desenvolvendo uma nova versão, "iniciada do zero", melhor estruturada e documentada.

Para que o iVProg pudesse ser utilizado como *applet* foi necessário um grande esforço para reduzir o tamanho do código do Alice, de 140Mb para cerca de 1,3Mb. Para isso preservamos apenas as funcionalidades relacionadas a criação de código fonte, as funcionalidades relacionadas a animações em 3D foram removidas por não fazer parte do projeto didático que tínhamos para o sistema.

A integração do iVProg ao SGC foi possível através de um módulo para o Moodle, que possibilita sua integração com *Módulos de Aprendizagem Interativa (iMA)* [Dalmon et al. 2011b], que é o caso do iVProg. O módulo em questão é o *iTarefa* [Rodrigues 2010], ilustrado na figura 1 com o iVProg integrado ao Moodle.



Figura 1. Tela do iVProg vinculado ao Moodle através do plugin iTarefa.

As funcionalidades relacionadas a criação de código fonte permitem que os aprendizes representem algoritmos visualmente, utilizando blocos representando os laços de repetição, estruturas de decisão, variáveis e funções, dentre outros. Estes blocos tem estrutura sintática análoga a de uma linguagem de programação estruturada tradicional, como a linguagem *C*. Na versão atual, por herança do Alice, o iVProg utiliza o método de interação de arrastar-e-soltar (*drag&drop*) para montar o algoritmo

(o novo permitirá também pegar-e-colar). Portanto, o aprendiz precisa apenas escolher o tipo de bloco (*se/senão*, *enquanto*, etc) e levá-lo até a posição desejada. Depois de montar o código utilizando esses blocos o aprendiz pode “compilar/rodar” seu programa utilizando o interpretador do iVProg, que o transformará numa representação interna do sistema e o executará, mostrando eventuais saídas de dados do algoritmo implementado. Para isso basta clicar no botão “Rode”, presente na figura 2 em modo desabilitado (pois o sistema esta em modo de execução de código).

Uma vantagem desse modelo está em reduzir significativamente a preocupação do aprendiz com detalhes da sintaxe ou com a configuração do ambiente de desenvolvimento-compilação. Desse modo, seu foco pode ser direcionado ao entendimento do problema e desenvolvimento do algoritmo que o soluciona.

A figura 2 exibe a tela do iVProg com um exemplo de código em execução.

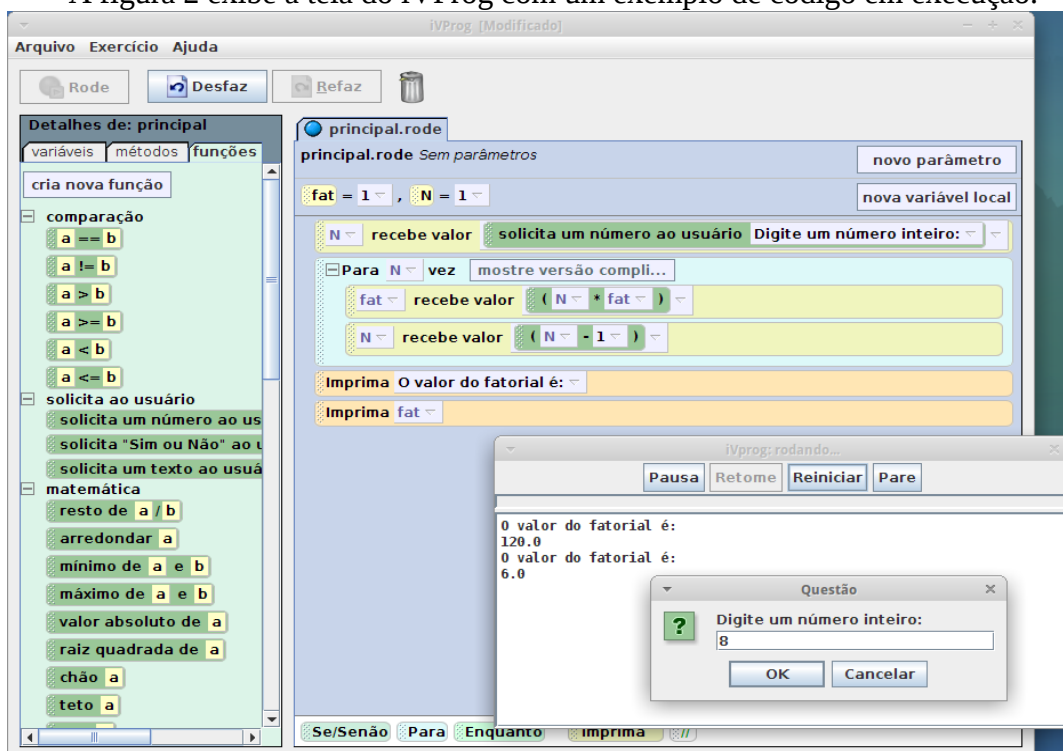


Figura 2. Tela do iVProg com código para cálculo de fatorial em execução.

2. Aplicação em Sala de Aula

O iVProg já foi utilizado em dois cursos de Introdução a Programação, ministrado para duas turmas de Licenciatura em Matemática, em 2010 (*curso II*) e 2011 (*curso III*). Os cursos tiveram duração de 18 semanas e foram ministrados de maneira mista, ou seja, além das atividades e aulas presenciais, os estudantes tinham muitas atividades a serem entregues semanalmente a distância pelo Moodle. As aulas presenciais ocorreram dentro de um laboratório de informática com, no máximo, dois estudantes por computador. Para efeitos de comparação, apresentaremos os dados do *curso I*, ministrado em 2005 pelo mesmo professor, também em laboratório e utilizando sistema interativo para programação com apoio Web. Entretanto, no curso I não foi empregado qualquer ambiente de programação visual, sendo utilizado como iMA o sistema livre iCG (disponível gratuitamente em <http://www.matematica.br/icg>). Nos três cursos a linguagem C foi utilizada, sendo que nos cursos I e II ela foi utilizada a partir da semana

6, enquanto no curso III foi utilizada desde o começo, introduzida em paralelo com o iVProg.

A proposta do cursos II e III era analisar se a programação visual com o iVProg facilitaria a compreensão do conceito de algoritmo e a introdução de linguagem C.

Nas 6 primeiras semanas do curso I utilizou-se apenas o iCG, a partir do qual os conceitos básicos de linguagem de programação foram deduzidos (como variáveis, expressões e desvios) e apresentados na sintaxe da linguagem C. A partir da semana 7, passou-se a utilizar apenas C, com atividades desenvolvidas também em laboratório.

No curso II, nas suas 6 primeiras semanas utilizou-se apenas o iVProg e depois mudou-se para o ambiente tradicional de programação em C, sempre em laboratório. Já no curso III, ambas as ferramentas foram utilizadas paralelamente nas 6 primeiras semanas de aula, sendo que primeiro eram resolvidos problemas utilizando o iVProg e depois, com o mesmo problema, utilizando a linguagem C.

No total, no curso I foram resolvidos 8 exercícios do iCG e 15 em C no laboratório e depois 3 exercícios mais complexos, que exigiam modelagem, foram resolvidos fora das aulas. No curso II foram resolvidos 20 exercícios usando o iVProg e 33 exercícios usando a linguagem C, em sua maioria no laboratório. Finalmente, no curso III foram resolvidos 25 problemas usando tanto C quanto iVProg e apenas 4 exercícios usando apenas a linguagem C. Houveram mais exercícios em C do que em iVProg porque o iVProg ainda não possui suporte a vetores e matrizes.

O desempenho dos estudantes foram baseadas nos exercícios resolvidos (C e não-C) em laboratório e nas avaliações 1 (P1) e 2 (P2). A tabela seguinte exibe os dados referentes a frequência dos alunos nos cursos, notas em P1 e P2 e notas nas listas de exercícios C e não-C (não-C significa usando o iCG para o curso I e o iVProg para os cursos II e III).

Tabela 1 – Comparação entre os anos de 2005 sem programação visual e 2010 e 2011 com programação visual

	Curso I (2005)		Curso II (2010)		Curso III (2011)	
	Média	Variância	Média	Variância	Média	Variância
frequência	56,48%	0,109	72,92%	0,072	71,02%	0,061
P1	3,28	8,3	5,59	8,7	5,85	5,64
P2	4,33	17,02	3,47	4,92	6,36	7,29
não-C	7	20,5	9,26	0,35	9,59	0,65
C	4,28	18,7	8,51	0,43	9,08	0,74
Média final	4,06	10,44	4,59	6,31	5,24	6,19

A partir da tabela 1 podemos observar um interessante aumento na frequência às aulas nos cursos II e III em relação ao curso I. Além disso as médias em avaliações e exercícios (C e não-C) foram maiores com o uso do iVProg, inclusive as médias finais. Esses números permite conjecturar que o iVProg teve impacto positivo na motivação e no aprendizado dos alunos nos cursos II e III, sugerindo ainda que a transição para a linguagem C é melhor realizada quando o iVProg e a linguagem C são apresentados em paralelo.

Atualmente o sistema iVProg encontra-se em fase de reconstrução, motivado por

diversas necessidades e ideias de melhorias, sintetizadas na próxima seção.

3. Reconstrução da Ferramenta

A ferramenta está em fase de refatoração completa. A principal razão para esta reconstrução está relacionado a adequação do iVProg com a arquitetura da Linha de Produto de Software (LPS) que está em desenvolvimento pelo grupo de pesquisa *LInE* (Laboratório de Informática na Educação). Para a criação da LPS foi desenvolvido um arcabouço de aplicação para iMA [Dalmon et al. 2011a]. Além dessa adaptação de arquitetura, são fatores que apoiaram tal decisão de reconstrução: i) adição de novo método interação, do tipo “Clique-pegar & Clique-cola”; ii) criação de tutor inteligente; e iii) adição de avaliação automática. Além desses fatores principais havia demandas pedagógicas, como o suporte a vetores e a matrizes, além da necessidade de uma interface flexível e adaptativa para ser, eventualmente, utilizada com estudantes do ensino médio.

A integração com o arcabouço iMA será realizada para colocar o iVProg dentro do protocolo e nos mesmos moldes da linha de produto de software caracterizada pelos iMA. O novo método de interação visa melhorar a usabilidade da ferramenta, pois pesquisadores da área de Interação Humano Computador apontam que, em relação ao uso de *mouse*, existe maior conforto e preferência para a utilização do método “Clique-pegar & Clique-cola” de que o “arrastar-e-soltar” (*drag-and-drop*) [Inkpen 2001]. No novo iVProg ambas as formas de interação serão permitidas, cabendo ao usuário escolher quando usar uma ou outra. Com o *tutor inteligente* e a *avaliação automática*, pretendemos melhorar a retroação (*feedback*) nas tarefas realizadas, o que geralmente melhora significativamente a motivação e satisfação dos aprendizes [Hara e Kling 2001]. A previsão para conclusão da primeira versão do novo iVProg é início de 2012.

4. Conclusão

Este trabalho apresentou uma ferramenta de programação visual, o iVProg, cujo objetivo primário é melhorar o processo de ensino-aprendizagem de algoritmos e programação. Em estudos de caso iniciais a ferramenta reduziu a sobrecarga cognitiva referente a sintaxe de linguagem e aprendizado de ambientes de programação, o que permite que os aprendizes se concentrem mais nos problemas a serem resolvidos.

Os próximos passos no projeto consiste em finalizar o núcleo do novo iVProg e depois iniciar os trabalhos no novo modelo de interface flexível e nos tutores (avaliação automática e tutoria inteligente). Estas tarefas tem impacto para professores, para aprendizes e também para os programadores-desenvolvedores do projeto. Os aprendizes poderão receber instantaneamente informações importantes sobre suas tentativas de soluções para problemas propostos. Os professores podem mais rapidamente tomar conhecimento sobre o desempenho de seus estudantes e qualidade das atividades que propõem. Já os programadores do projeto terão um código que poderá mais facilmente ser mantido e estendido.

5. Agradecimentos

O grupo de pesquisa agradece aos financiamentos parciais do CNPq (550422/2011-6), FAPESP (2011/10926-2) e à CAPES pela bolsa institucional de mestrado.

6. Referências

- Caspersen, M. (2009) "STREAM: A first programming process," *ACM Transactions on Computing Education*. Volume 9, Número 1.
- Conway, M., J. e Pausch, R. (1997) "Alice: Easy to Learn Interactive 3D Graphics", *Computer & Graphics*, Volum 31, Número 3, páginas 58-59.
- Dalmon, D., Brandão, A. e Isotani, S. (2011a) " Work in Progress - A Framework for Building Interactive Learning Modules". In: *Frontiers in Education Conference*, páginas 11-12.
- Dalmon, D. L., Tanbellini, M. J. G. S., Eisenmann, A., Nascimento, M., Rodrigues, P. A., Isotani, S., Brandão, A. A. F. e Brandão, L. O. (2011b) "Interactive learning modules in engineering education and as a motivational tool for middle and high school students", In: *Proceedings of International Symposium on Engineering Education – IGIP*.
- Hara, B. N., Kling, R. (2000) "Student Distress in Web-based Distance Education Course". In: *Information, Communication & Society*. Volume 3, Assunto 4, páginas 557-579.
- Inkpen, K. M. (2001) "Drag-and-drop versus point-and-click mouse interaction styles for children". In: *ACM Transactions on Computer-Human Interaction*. Volume 8, Número 1, páginas 1-33.
- Kamiya, R. R., Brandão, L. O. (2009) "iVProg - um sistema para introdução à Programação através de um modelo Visual na Internet," In: *Anais do Simpósio Brasileiro de Informática na Educação*.
- Rodrigues, P. A., Brandão, L. O., Brandão, A. A. F., (2010) "Interactive Assignment: a Moodle component to enrich the learning process". In: *Proceedings of Frontiers in Education*, páginas T4F-1 - T4F-6.