

# RHODES 2.0: Software Educacional para o Ensino de Programação Linear

Wanderley de Souza Alencar<sup>1</sup>, Bruno Moraes Rocha<sup>2</sup>, Emília Alves Nogueira<sup>2</sup>, Rodrigo Fideles Fernandes<sup>3</sup>, Fabrizzio Alphonsus Alves de Melo Nunes Soares<sup>1</sup>

<sup>1</sup>Instituto de Informática – Universidade Federal de Goiás (UFG)  
Caixa Postal 131 – 74.001-970 – Goiânia – GO – Brazil

<sup>2</sup>Departamento de Ciência da Computação – Universidade Federal de Goiás (UFG)  
Caixa Postal 01 – 75.801-615 – Jataí – GO – Brazil

<sup>3</sup>Departamento de Educação – Pontifícia Universidade Católica de Goiás (PUC-GO)  
Caixa Postal 86 – 74.605-020 – Goiânia – Brazil

{wanderleyalencar, fabrizzio}@inf.ufg.br

{ufg3runo, emiliacdc, rodrigo.fidelis}@hotmail.com

**Abstract.** *This paper presents an educational software, free and multiplatform, designed with objective of the collaborating with professors in the teaching Operations Research, since it's present in many undergraduate courses in Brazilian universities. The software - called RHODES, version 2.0 - helps teaching the Simplex Method, providing functionality to algebraic formulation of problems, presentation graphics in 2D/3D and Simplex tableaus, the possibility of setting up the database with example problems, among others. Functional evaluations performed by the authors demonstrated its usefulness as an auxiliary tool for improving the teaching-learning process in the classroom or in outside of the class activities.*

**Resumo.** *Este trabalho apresenta um software educacional, livre e multiplataforma, concebido com o objetivo de colaborar com os docentes no magistério da disciplina Pesquisa Operacional presente nos mais variados cursos superiores brasileiros. O software – nomeado por RHODES 2.0 – auxilia o ensino do Método Simplex, oferecendo funcionalidades para formulação algébrica de problemas, apresentação de gráficos em 2D/3D e de tableaus Simplex, bem como a possibilidade de constituição de base de dados com problemas-exemplo, dentre outras. Avaliações funcionais realizada pelos autores evidenciaram sua utilidade como ferramenta auxiliar para a melhoria do processo de ensino-aprendizagem em sala de aula ou extraclasse.*

## 1. Introdução

A sociedade do século XXI pode ser caracterizada pela extrema velocidade com que ocorrem a geração, divulgação e uso tecnológico dos novos conhecimentos nas mais diversas áreas da atividade humana. Uma das consequências desta dinâmica tem sido a crescente necessidade de refletir a respeito de (novas) metodologias a serem empregadas para a concepção, planejamento e condução dos processos de ensino-aprendizagem, seja no espaço formal das Instituições de Ensino (IEs) ou nos espaços informais, e extremamente capilarizados, do convívio social vivenciado pelos indivíduos.

Especialmente no contexto escolar, independente do *nível de ensino*<sup>1</sup>, o *modelo tradicional de educação*<sup>2</sup> tem sido criticado por diversos pesquisadores por não atender às necessidades da sociedade contemporânea e algumas propostas metodológicas de enfrentamento têm sido concebidas ([Ausubel *et al* 1980], [Novak 1998] e [Gardner 2000]). Algumas discutem a utilização das novas Tecnologias de Informação e Comunicação (TICs) como ferramenta auxiliar para o estabelecimento de um processo de ensino-aprendizagem mais dinâmico, interativo e empático aos estudantes e professores ([Pais 2002], [Valente 2003] e [Leite 2008]).

As sofisticadas capacidades de visualização/interação ofertadas pelas TICs podem estimular os estudantes para a adoção de uma nova postura em relação ao processo de ensino-aprendizagem no âmbito escolar: uma postura mais (pro)ativa, participativa e engajada na construção de seu próprio conhecimento. Entretanto, a simples disponibilidade de recursos tecnológicos sofisticados não permite que se assegure a melhoria dos processos de ensino-aprendizagem e, portanto, da educação, como evidenciado por [Cardoso 2010 p. 26]:

A depender da forma como seja planejada a utilização dos novos recursos tecnológicos em um ambiente educacional, possibilita enriquecer e ampliar as condições e as chances de construção do conhecimento pelo aluno e a consequente melhoria no aprendizado. Ao adotar diferentes métodos e abordagens de ensino com novos recursos tecnológicos, complementando os modelos convencionais já amplamente utilizados no cotidiano dos professores, pode-se flexibilizar a maneira como ocorrem os momentos vividos presencialmente em sala de aula com os de aprendizagem virtual, de forma integrada e alternada.

Em consonância com diversas propostas de complementariedade entre os modelos convencionais de ensino e aqueles que empregam TICs, especialmente *softwares educacionais* como os propostos por ([Reis *et al* 2009], [Rodger *et al* 2009], [Klock *et al* 2010], [Vieira *et al* 2010] e [Madsen e Adamatti 2011]), é que os autores deste trabalho propõem a construção de um *software educacional livre*<sup>3</sup>, em português, para ser utilizado como ferramenta auxiliar para o ensino da *Programação Linear (PL)* – especificamente do clássico método de resolução denominado *Método Simplex* – constante em muitos currículos plenos de cursos superiores no Brasil e no mundo, conforme ([SBC 2005] e ([ACM e IEEE 2008]), por exemplo.

Este trabalho está assim organizado: a Seção 2 abordará a programação linear e como se dá a prática de seu ensino nos cursos de nível superior brasileiros. A Seção 3 apresentará o *software educacional* proposto – denominado RHODES, versão 2.0 – para o ensino de PL. A Seção 4 apresenta os resultados obtidos na avaliação funcional do *software* e, finalmente, a Seção 5 apresenta as conclusões e possíveis trabalhos futuros, alguns dos quais já em processo de desenvolvimento pelos autores.

<sup>1</sup> No Brasil, a Lei de Diretrizes e Bases da Educação Nacional (LDB) – Lei nº 9394, de 20/12/1996 – define que a *educação escolar* é composta da educação básica e da educação superior (Art. 21, *caput*). A primeira é subdividida em educação infantil, ensino fundamental e ensino médio (Art. 21, inciso I). A segunda está subdividida em cursos sequenciais, de graduação, de pós-graduação e de extensão (Art. 44).

<sup>2</sup> Ou *Educação Depositária*: neste modelo o professor é o centro do processo de ensino-aprendizagem, detentor do conhecimento e responsável por sua transmissão. Ao estudante (considerado aqui simples *aprendiz*) cabe recebê-lo e armazená-lo, normalmente por meio de leituras repetitivas, pela cópia e resolução de exercícios de *fixação* ([Mizukami 1986], [Libâneo 1992] e [Gadotti 1995]).

<sup>3</sup> *Software livre* (ou *Free Software*) para a *Free Software Foundation* (<http://www.fsf.org>), é aquele que pode ser usado, copiado, estudado, modificado e redistribuído sem nenhuma forma de restrição, sendo acompanhado por uma licença de *software livre* como a GPL (*General Public License*), por exemplo.

## 2. A Programação Linear e seu Ensino em Cursos Superiores Brasileiros

A disciplina nomeada de *Pesquisa Operacional (PO)*<sup>1</sup> consta como integrante do currículo pleno dos mais variados cursos superiores, como: Administração, Ciência da Computação, Economia, Engenharia da Computação, Física, Logística, Matemática e Química, dentre outros. Em alguns destes sua integralização é obrigatória e, em outros, apesar de opcional, é fortemente recomendada como necessária à plena formação do acadêmico. Tipicamente nas IEs brasileiras sua carga horária varia entre 32 a 80 horas.

Especialmente nos cursos da área de Computação e Informática, a PO consta das recomendações curriculares para os cursos de Ciência da Computação ([SBC 2005], [ACM e IEEE 2008]) e Engenharia da Computação [SBC 2005], por exemplo. Os objetivos, carga horária total, conteúdo e metodologia de ensino empregada podem variar entre as instituições, e de curso para curso numa instituição. Apesar destas especificidades, o tema da programação linear está presente na totalidade das ementas da disciplina, posto que ele é basilar para boa compreensão da área de PO e, portanto, de conhecimento essencial para todos aqueles estudantes.

Normalmente o tema da PL contempla pelo menos os seguintes tópicos: (1) estudo/elaboração de modelos lineares de otimização para problemas – reais e/ou fictícios; (2) estudo dos algoritmos *Simplex* e *Simplex Revisado*; (3) da dualidade; (4) do algoritmo primal-dual; (5) realização de análise de sensibilidade; e (6) estudo da programação linear inteira. Uma abordagem detalhada a respeito destes assuntos pode ser obtida em [Bazaraa 2009], dentre diversos outros.

Num grande número de universidades brasileiras, a principal metodologia de ensino empregada para abordar a PL pode ser sintetizada por um processo educativo que envolve a ocorrência de quatro fases sequenciais, com algum nível de imbricamento entre elas: (1) apresentação da fundamentação teórica do assunto; (2) resolução de problemas-modelo em sala de aula; (3) submissão de lista de exercícios contemplando problemas a serem resolvidos pelos estudantes, seja em sala de aula ou extraclasse; e, finalmente, (4) resolução, pelo professor, de alguns exercícios-chave selecionados com o objetivo de proporcionar a fixação dos conceitos teóricos estudados.

Durante a primeira fase – *apresentação da fundamentação teórica* – os principais conceitos da PL são geralmente apresentados pelo professor por meio de aulas expositivas e/ou dialogadas, por vezes amparadas pelo uso de recursos multimídia como a projeção de *slides* (elaborados por meio de algum programa de autoria de conteúdos), sendo cópia destes fornecida, prévia ou posteriormente, aos estudantes.

A vivência dos autores em PO tem evidenciado que, apesar deste aparato tecnológico de suporte usado nesta etapa, é recorrente que os estudantes universitários apresentem dificuldades para a necessária compreensão, abrangente e plena, dos fundamentos da PL [Stevens and Palocsay 2004]. Três possíveis razões são: (1) as

---

<sup>1</sup> Outras denominações costumeiras nos currículos plenos dos cursos superiores das universidades brasileiras incluem: programação matemática, programação linear, otimização de sistemas, matemática computacional, otimização combinatória, dentre diversos outros. Internacionalmente: (*introduction to linear programming, operations research, mathematical programming, etc.*)

características<sup>1</sup> da formação básica da maior parte dos estudantes brasileiros fazem com que haja enorme dificuldade para a internalização de conceitos abstratos típicos da PL; (2) inadequado arcabouço matemático, advindo da deficiente formação educacional dos níveis de ensino fundamental e médio, muitas vezes impedem os estudantes de compreender operações matemáticas que deveriam ser consideradas triviais no nível superior de ensino; (3) uso de metodologias de ensino inadequadas por muitos professores da disciplina e do ensino superior de forma geral, calcadas na *educação tradicional* previamente aludida.

A disponibilidade de um *software educacional* com capacidade de proporcionar a *visualização/interação* com os conceitos abstratos sendo abordados, que facilite o acompanhamento e compreensão de cada um dos raciocínios necessários para a correta modelagem e resolução de PLs pode contribuir para o aprimoramento do processo de ensino-aprendizagem ao colaborar para que se minimizem as dificuldades (1) e (2) *supra* identificadas. O combate à dificuldade (3) exige ações coordenadas por parte dos professores, das IEs e do Estado para empregar metodologias que impeçam os estudantes de interagirem de forma superficial e mecanicista com os conteúdos.

Com o propósito de contribuir para o aprendizado de PL na segunda fase – *a resolução de problemas-exemplo em sala de aula* – é possível o emprego de *softwares aplicativos*<sup>2</sup> destinados à modelagem e resolução de problemas lineares. Alguns sistemas comerciais disponíveis são: Analytica, Cplex, GUROBI, KNITRO, LINDO e LINGO. Entretanto, o uso destes sistemas apresenta pelo menos três inconvenientes: (1) elevado custo para o licenciamento de cópias para as IEs brasileiras<sup>3</sup>; (2) a impossibilidade de conhecer seu funcionamento interno, por serem eles fornecidos como *caixas pretas* e originalmente são destinados ao uso profissional, e não acadêmico, protegendo os direitos de propriedade intelectual de seus fabricantes; e (3) usabilidade e flexibilidade inadequadas às necessidades didático-pedagógicas típicas do ambiente escolar, em que deve prevalecer a exploração, e não a obtenção do resultado final no “*menor tempo possível*”, objetivo normalmente perseguido em produtos comerciais.

O segmento dos *softwares livres* oferecem soluções sob a forma de *solvers* e/ou *bibliotecas de desenvolvimento*, alguns exemplares são: ABACUS, CBC, CMPL, GAMS, LP\_SOLVER<sup>4</sup>, dentre outros. O maior inconveniente é que o uso destes *softwares* normalmente requer considerável conhecimento de programação de computadores, o que não pode ser exigido de todo professor de PL, notadamente daqueles não vinculados a cursos de Informática e Computação.

<sup>1</sup> Os estudantes normalmente são oriundos de instituições de ensino em que se aplica a *Pedagogia Depositária* (ou *modelo tradicional de ensino*) previamente referida neste texto.

<sup>2</sup> Este tipo de *software* é normalmente conhecido pelo nome, em inglês, de *solver* – resolvidor, em livre tradução. Apesar disso, alguns apresentam capacidades adicionais que não vinculadas diretamente apenas à obtenção das soluções, se existentes, para um problema que lhes seja submetido.

<sup>3</sup> Apesar de haver aqueles *softwares* que possibilitam o uso acadêmico, desde que observadas algumas *restrições de uso*, cujas mais comuns (1) a limitação quanto ao número/tamanho dos problemas que podem ser modelados e/ou resolvidos e (2) o tempo de validade da licença (tipicamente um mês).

<sup>4</sup> Informações detalhadas sobre diversos *solvers* estão disponíveis em <<http://www.aimms.com/aimms/solvers>> e <<http://www.dmoz.org/Science/Math/Combinatorics/Software>>.

A proposta dos autores é o desenvolvimento de *software educacional* projetado especificamente para as necessidades didáticas do típico professor de PL, em que sejam privilegiadas a usabilidade dos usuários e a fidelidade com os conceitos teóricos sendo explorados. A Seção 3 mostra, como já mencionado, os detalhes deste *software*.

Para a terceira e quarta etapas do processo educativo – *aplicação de listas de exercícios e posterior resolução pelo professor de alguns exercícios-chave selecionados* – a disponibilidade de um *software educacional* que possibilite a integração, passo a passo no processo resolutivo do problema, entre a resolução algébrica e a visualização gráfica (para problemas com até três variáveis de decisão), permite maior facilidade discussão e compreensão entre professores e estudantes. Além disso, estudantes poderão realizar, mais facilmente, experimentações extraclasse que os conduzam à abrangente e plena assimilação e internalização do *Método Simplex*.

Há relato favoráveis do emprego de *software educacional* em PL como as propostas apresentadas por ([Lucindo 2006], [Lazaridis *et al* 2007] e [Santos 2012])). Seguindo esta abordagem, a iniciativa dos autores é apresentada na próxima seção.

### 3. O *Software Educacional RHODES* para Ensino de Programação Linear

O *software educacional RHODES* é livre e tem seu código-fonte escrito em Português. Seu desenvolvimento acontece desde o início de 2011, e envolve uma equipe multidisciplinar com docentes das áreas de Pesquisa Operacional, Ciência da Computação (CdC) e de Educação, além de acadêmicos do curso de CdC. A versão atual é a 2.0, com perspectiva para o lançamento de uma nova versão anualmente.

*RHODES* destina-se ao ensino do *Método Simplex* para resolução de problemas de programação linear (PPLs) e foi concebido com o objetivo de estimular e auxiliar a aprendizagem dos alunos, instigando a reflexão e a experimentação autônoma durante a construção de seu conhecimento. O *software* utiliza uma interface gráfica dotada de múltiplas janelas para possibilitar alto nível de interação com os conceitos sendo abordados, bem como proporcionar visão holística deles – o que tende a beneficiar a aprendizagem devido ao maior envolvimento dos estudantes [Lazaridis *et al* 2007].

Por meio desta visão integrada e abrangente do processo modelagem/resolução de problemas, o professor passa a proceder como mediador entre o estudante e os conceitos da PL. Ele – estudante – consegue perceber mais facilmente a relação existente entre os conceitos abstratos e as operações concretas que, no método tradicional de ensino, eram examinados de forma isolada (resoluções algébrica e gráfica). Isto atua como estímulo à participação ativa na construção do conhecimento.

O *software* apresenta uma interface que tem sido considerada – nos testes funcionais realizados – suficientemente amigável, o que permite ao estudante facilidades para a manipulação simultânea do modelo algébrico (apresentado sob a forma de um *tableaux Simplex* em uma janela independente do sistema – a *janela de expressão*) e do modelo gráfico correspondente em 2D ou projeção 3D, de acordo com as características do problema. As Figuras 1 e 2 apresentam estes detalhes.

A resolução de um PPL modelado pode, por escolha do estudante, ser realizada num único passo ou na forma *passo a passo*, o que lhe permite a análise detalhada de

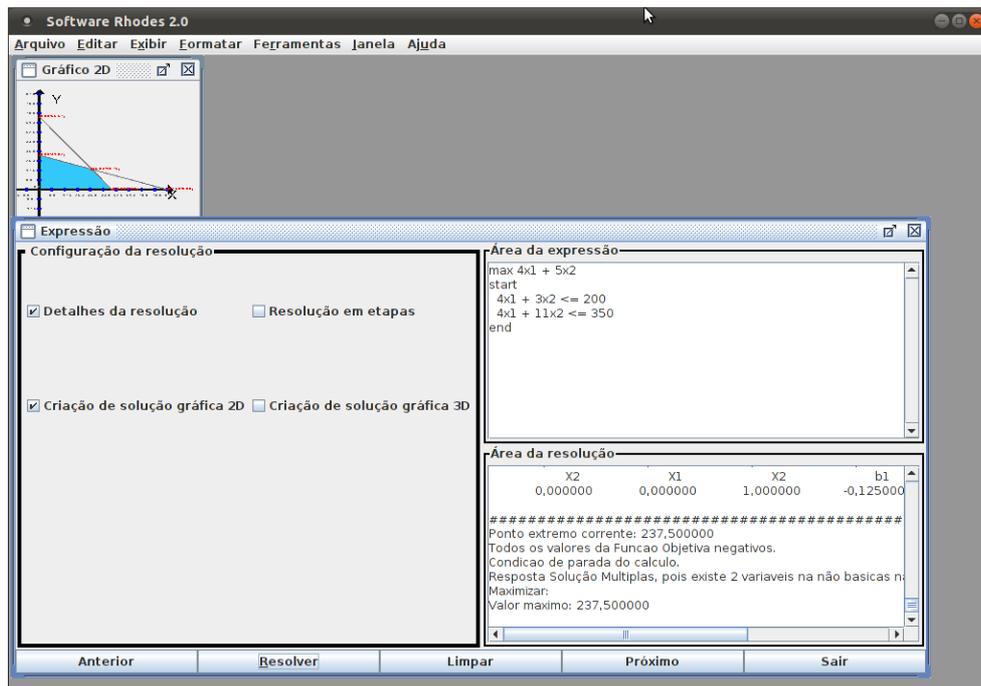


Figura 1. Janelas de configuração da solução de um PPL (com duas variáveis), de expressão (modelagem), de resoluções algébrica e gráfica em 2D.

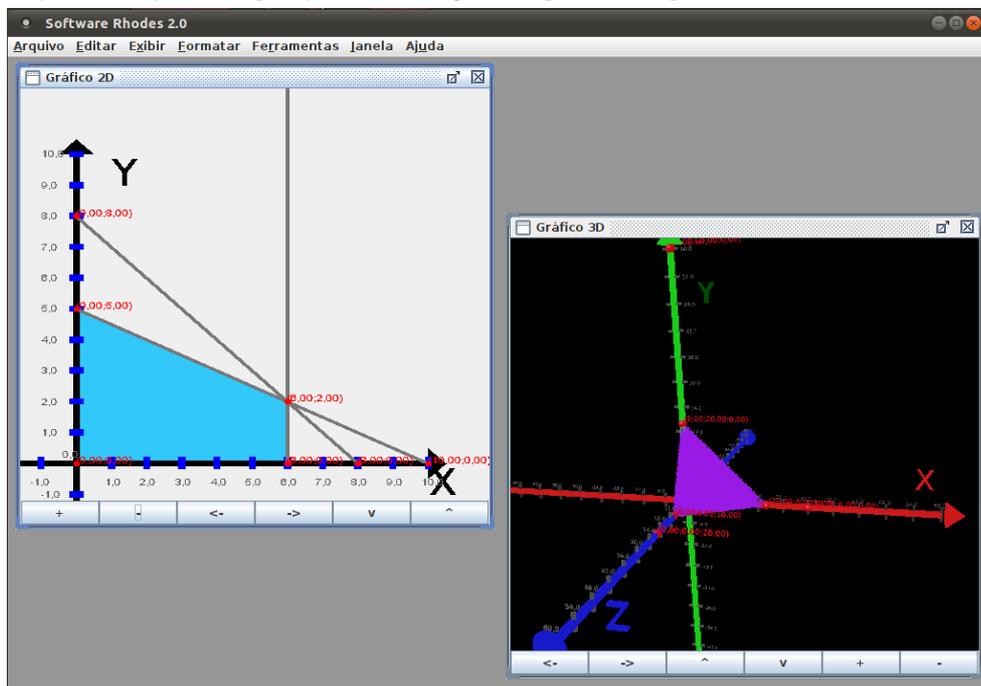


Figura 2. Exemplos de soluções de problemas de programação linear em duas e três dimensões, sendo a última apresentada sob a forma de projeção.

cada etapa envolvida na síntese da solução do problema, bem como o entendimento dos conceitos/operações necessárias para sua obtenção.

O fornecimento do modelo de PPL para o RHODES 2.0 pode ser realizado por meio da “importação” de modelos previamente padronizados, elaborados e armazenados em um simples arquivo do tipo *texto*, ou de forma interativa numa janela

destinada à modelagem (*janela de expressão do problema*). A Figura 1 ilustra o modelo interativo, onde se configura o modo de resolução.

Numa janela gráfica (2D ou projeção 3D) é possível ao estudante visualizar e manipular – com rotações, *zoom* e outros recursos – toda a fase de resolução do PPL, onde são destacados a região factível e seus pontos extremos (vértices do polítopo convexo gerado pela conjunção das restrições tecnológicas impostas ao problema). Outra funcionalidade é a possibilidade de criar inúmeros problemas/soluções, no qual permite ao usuário fazer comparações entre os resultados gráficos 2D e 3D, permitindo análises comparativas entre problemas, como ilustrado na Figura 3.

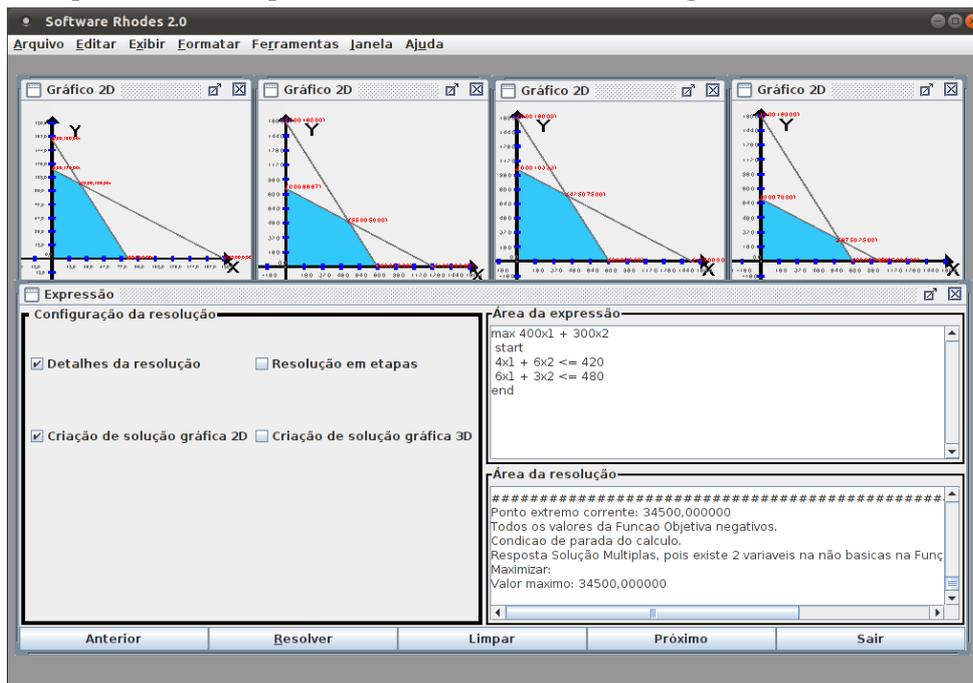


Figura 3. Comparação das soluções gráficas, em duas dimensões, de diversos problemas de programação linear.

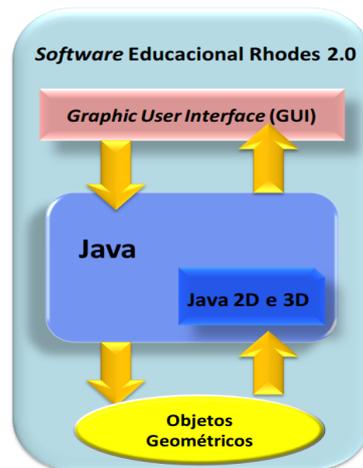
### 3.1. Implementação do RHODES 2.0

A implementação utiliza a linguagem Java para manipulação dos cálculos matemáticos para a resolução dos PPLs e as APIs (*Application Programming Interface*) Java 2D e 3D para a gestão dos objetos virtuais. O ambiente de desenvolvimento NetBeans IDE 7.2 ((IDE, *Integrated Development Environment*) foi escolhido e instalado sob sistema operacional Linux Ubuntu 11.04, entretanto o *software* é multiplataforma e portátil para outras distribuições Linux, para Windows e Mac OS. A estrutura do RHODES é representada na Figura 4, onde se destacam:

(1) Interface Gráfica do Usuário (*Graphic User Interface*): permite ao usuário fornecer os modelos algébricos, acompanhar a execução dos algoritmos de resolução (por meio dos *tableaus* apresentados em janela), interação/visualização dos gráficos 2D e/ou 3D da região de factibilidade do PPL;

(2) Motor Java com APIs Java 2D e 3D: gerencia os ambientes de manipulação dos modelos e da resolução dos problemas, e se comunica com as APIs Java 2D e 3D para a geração e apresentação de gráficos, assim como valida a navegação e interação direta com os objetos virtuais na camada dos objetos geométricos;

(3) **Objetos Geométricos:** Responsável por todas as transformações geométricas necessárias para a apresentação dos gráficos.



**Figura 4. Estrutura interna do RHODES 2.0: a interface gráfica de usuário, o motor Java e suas APIs e, finalmente, a manipulação de objetos geométricos.**

#### 4. Avaliação Funcional do RHODES 2.0

O RHODES 2.0 foi avaliado sob a ótica funcional proposta por [Nielsen 2000] por um grupo de dez acadêmicos de um curso de Bacharelado em Ciência da Computação aleatoriamente escolhidos de um grupo que contava com trinta e cinco estudantes que cursaram recentemente (no primeiro semestre letivo de 2012) a disciplina de PO/PL segundo a metodologia de ensino nomeada de *modelo tradicional*.

A avaliação consistiu da proposição aos estudantes que realizassem a resolução de exercícios envolvendo PPLs, sem a intervenção de nenhum instrutor ou realização de treinamento prévio sobre o funcionamento do *software*, exceto a indicação do local em que se encontravam armazenados os exercícios a serem resolvidos. Os PPLs resolvidos por diferentes estudantes eram também diferentes.

Ao final os estudantes responderam a um *Questionário de Avaliação de Uso*, que considerava as seguintes características: (1) nível de facilidade para a interatividade com o *software*; (2) nível de usabilidade do *software*; (3) nível em que o uso do *software* como ferramenta auxiliar, de forma geral, colaboraria para a compreensão dos conceitos e conhecimentos exigidos para a resolução dos PPLs; (4) nível em que o uso da apresentação gráfica da(s) solução(ões) em 2D e/ou 3D, o auxiliaria na compreensão conceitual do assunto; e, por fim, (5) o quanto o *software*, com seus exercícios resolvidos no ambiente proporcionado, o auxiliaria no estudo de forma autônoma.

Considerando as respostas *Muito Boa* e *Boa* como resultados positivos, obteve-se média geral de 90%, conforme mostra a Tabela 1, o que sinaliza que o *software* RHODES pode ser ferramenta auxiliar útil no processo de ensino-aprendizagem de PL. Os entrevistados relataram, informalmente, que consideravam que haveria maior facilidade para a compreensão de diversos conceitos – e exercícios – se contassem com a ferramenta quando cursaram a disciplina PO/PL. Os itens que se referem à usabilidade (item 2) e recursos gráficos em 2D/3D (item 4) atingiram 100% de positividade por meio da associação do conceito *Muito Boa*.

**Tabela 1. Resultados positivos, neutros e negativos de acordo com o questionário.**

Característica	Positivo (%)	Neutro (%)	Negativo (%)
1	90,00%	10,00%	0,00%
2	100,00%	0,00%	0,00%
3	80,00%	20,00%	0,00%
4	100,00%	0,00%	0,00%
5	80,00%	20,00%	0,00%
Média	90,00%	10,00%	0,00%

#### 4. Conclusões e Trabalhos Futuros

No momento, RHODES continua sendo testado em ensaios progressivamente mais rigorosos, com comparativos com mais de duzentos PPLs, em que o número de variáveis, tipo e quantidade de restrições envolvidas são variados. Os resultados obtidos são contrastados com o de outros *solvers* comercialmente disponíveis (LINDO e LINGO) e têm demonstrado a correção do sistema quanto à resolução de PPLs.

Para os PPLs que envolvem duas ou três variáveis de decisão, a visualização dos gráficos em 2D (polígonos) e projeção em 3D (poliedros convexos) de forma integrada aos modelos algébricos de resolução do problema tem se mostrado ferramenta didático-pedagógica conveniente, como descrito anteriormente, em testes de laboratório envolvendo estudantes. Isto tem motivado os autores à continuidade do projeto.

Estão previstas as seguintes extensões para o trabalho: (1) inclusão da possibilidade de impressão da resolução detalhada de um PPL, inclusive com a representação gráfica, se houver; (2) inclusão da tecnologia de Realidade Aumentada com o uso de óculos para visualização em 3D, o que propiciará experiência mais intensa; (3) possibilidade de realização de avaliação automática de conhecimentos; (4) elaboração de versão para *web*; e (5) incorporação de outros métodos de resolução.

#### Referências

- Ausubel, D. P., Novak, Joseph D. and Hanesian, H. (1980). "Psicologia educacional". Rio de Janeiro: Interamericana.
- Bazaraa, B. S. (2009). "Linear programming and network flows", 4th. ed., John Wiley and Sons.
- Cardoso, Antônio Luiz M. de S. (2010). "Construção e difusão colaborativa do Conhecimento: uma experiência construtivista de educação em um ambiente virtual de aprendizagem". 2010. 302f. Tese (Doutorado em Educação) – Faculdade de Educação, Universidade Federal da Bahia, Salvador.
- Costa, Cristina (2005). "Educação, imagem e mídias". São Paulo: Cortez, V. 12.
- Dávalos, Ricardo V. (2002). "Uma abordagem do ensino de pesquisa operacional baseada no uso de recursos computacionais". In: XXII Encontro Nacional de Engenharia de Produção, Curitiba.
- Gadotti, M. (1995). "Histórias das idéias pedagógicas", São Paulo: Ática.
- Gardner, H. (2000). "Inteligências múltiplas: a teoria na prática", Porto Alegre: Penso.
- Klock, C. E., Ribas, R. P. e Reis, A. I. (2010). "Karma: um ambiente de aprendizado para a síntese de funções booleanas". In: Revista Brasileira de Informática na Educação, Vol. 18, n. 2, pp. 34-42.

- Lazaridis V, Paparrizos K., Samaras N. and Sifaleras A. (2007). "Visual LingProg: a web based educational software form linear programming", *Computer Applications in Engineering Education*, Wiley Publications, Vol. 15, No. 1, pp. 1-14.
- Libâneo, J. C. (1992). "Democratização da escola pública: a pedagogia crítico-social dos conteúdos", São Paulo: Loyola, 1992.
- Leite, Lígia S. (2008). "Mídia e a perspectiva da tecnologia educacional no processo pedagógico contemporâneo". In: Freire, Wendel (Org.) *Tecnologias e Educação: as Mídias na Prática Docente*. Rio de Janeiro: Wak.
- Lucindo, Renato P. F. L, Witt, Thiago M., Campos, Cássio P. de e Gruber, Aritanan B. G. (2006). "jOptimum: um sistema de otimização". In: XIV Workshop sobre Educação em Computação (XXVI Congresso da SBC), Campo Grande, p. 158-167.
- Madsen, Carlos Alberto B. C. W. e Adamatti, Diana Francisca. (2011). "NeuroFURG: uma ferramenta de apoio ao ensino de redes neurais artificiais". *Revista Brasileira de Informática na Educação*, Vol. 19, n. 2. pp. 14-24.
- Mizukami, M. G. N. (1996). "Ensino: as abordagens do processo", São Paulo: EPU.
- Novak, Joseph D. (1998). "Creating and using knowledge: concept maps as facilitative tools for schools and corporations", Mahwah: Lawrence Erlbaum & Assoc.
- Nielsen, Jakob (2000). "Why you only need to test with 5 users". Jakob Nielsen's Alertbox. Disponível em <<http://www.useit.com/alertbox/20000319.html>>.
- Pais, Luiz Carlos (2002). "Educação escolar e as tecnologias da informática". Belo Horizonte: Autêntica.
- Reis, Fabrício P., Júnior, Paulo Afonso P. e Costa, Heitor Augustus, X. (2009). "TBC-SO/WEB: Um software educacional para o ensino de políticas de escalonamento de processos e alocação de memória em sistemas operacionais", In: XX Simpósio Brasileiro de Informática na Educação (SBIE 2009), Florianópolis.
- Rodger, Susan H., Wiebe, E., Lee, Kyung M., Morgan, C., Omar, K., and Su, J. (2009). "Increasing engagement in automata theory with JFLAP". In: Fourtieth SIGCSE Technical Symposium on Computer Science Education, pp. 403-407.
- Santos, Maurício P. dos (2012). "Software aplicativo para programação linear". Disponível em <<http://www.mpsantos.com.br/#soft>>. Acesso em 12 ago 2012.
- Sociedade Brasileira de Computação (2005). "Currículo de referência da SBC para os cursos de graduação em bacharelado em Ciência da Computação e Engenharia da Computação". Disponível em <<http://www.sbc.org>>. Acesso em 12 ago 2012.
- Stevens, Scott P. and Palocsay, Susan W. (2004). "A translation approach to teaching linear programming formulation". In: *Informatics Transac. on Education*, 4:3, pp. 38- 54.
- Valente, José Armando (2003). "O uso inteligente do computador na educação". In: *Pátio - Revista Pedagógica*, Vol. 1, n. 1, Artes Médicas Sul, pp. 19-21.
- Vieira, P. V., Raabe, A. L. e Zeferino, C. A. (2010). "Bípede - Ambiente de desenvolvimento integrado para a arquitetura dos processadores BIP. *Revista Brasileira de Informática na Educação*, Vol. 18, n. 1, pp. 32 -43.