

# Um *framework* construtivista no aprendizado de Sistemas Operacionais – uma proposta pedagógica com o uso do simulador SOsim

Francis Berenger Machado<sup>1</sup>, Luiz Paulo Maia<sup>2</sup>

<sup>1</sup>Depto. de Informática – Pontifícia Universidade Católica do Rio de Janeiro (PUC-Rio)  
Rua Marquês de São Vicente, 225 – Prédio do RDC – 4º andar – Gávea – 22453-900 –  
Rio de Janeiro – RJ – Brazil

<sup>2</sup>Núcleo de Computação Eletrônico – Universidade Federal do Rio de Janeiro  
Caixa Postal 2.324 – 20001-970 – Rio de Janeiro – RJ – Brasil

berenger@pobox.com, lpmaia@training.com.br

*Abstract. The traditional methodology that has been used to teach Operating Systems has not been enough for most of the students to learn sufficiently the concepts and mechanisms of the subject in this program. This paper shows a pedagogical proposal, which is based on constructive ideas to make the learning process of Operating Systems more effective and interesting. The framework presented uses the SOsim graphical simulator as a supporting tool for the constructive model, so as to make it possible to bring together both the learning process and practicing of theories learned in the classroom.*

*Resumo. A abordagem tradicional adotada para o ensino de Sistemas Operacionais não parece ser suficiente para que a maioria dos alunos tenha um aprendizado preciso dos conceitos e mecanismos que fazem parte do currículo da disciplina. O presente trabalho apresenta uma proposta pedagógica com base nas idéias construtivistas de forma a tornar o aprendizado de Sistemas Operacionais mais efetivo e interessante. O framework apresentado utiliza o simulador gráfico SOsim como ferramenta de suporte ao modelo construtivista, possibilitando a criação de um ambiente híbrido de ensino-aprendizado e a experimentação das teorias apresentadas em sala de aula.*

## 1. Introdução

Sistemas Operacionais é uma disciplina importante e obrigatória nos cursos de Ciência da Computação, Sistemas de Informação e Engenharia de Computação [DC-MEC 2004]. O currículo da disciplina envolve teoria, abstração e projeto, abordando conceitos e mecanismos um tanto complexos. A experiência de professores e alunos tem mostrado como é grande a dificuldade em ensinar e compreender os conceitos teóricos, aplicação prática e técnicas apresentadas. Ao contrário de outras disciplinas da área da computação, Sistemas Operacionais é um assunto que não possui uma estrutura linear, de forma que a docência possa ser realizada em estágios sequenciais e evolutivos. Conforme Maziero (2002), “uma das principais características da disciplina de Sistemas

Operacionais é a relativa dificuldade em definir um seqüenciamento didático claro entre seus diferentes tópicos”.

O modelo tradicional de aula em que o professor segue uma bibliografia, elabora transparências para suas aulas e aplica alguns exercícios teóricos, não parece suficiente para que a maioria dos alunos tenha uma compreensão precisa do que está sendo ensinado. O problema está tanto no modelo de ensino como na falta de ferramentas capazes de traduzir para a realidade os conceitos teóricos apresentados. Além disso, poucos laboratórios práticos são implementados, tornando as aulas extremamente teóricas e desmotivantes para os alunos. Na verdade, a falta de prática distancia o aluno do objeto em estudo, dificultando a real compreensão dos conceitos e mecanismos envolvidos.

A dificuldade no ensino de Sistemas Operacionais já vem sendo discutida há algum tempo por pesquisadores como Downey (1999) e Jones e Newman (2002). Uma alternativa que tem sido bastante explorada com alguns resultados positivos é a utilização do método construtivista no ensino da disciplina. O uso do construtivismo nas ciências da computação é recente, apesar de ser bastante difundido na área da matemática [Ben-Ari 1998]. No Brasil, existem poucos trabalhos acadêmicos publicados sobre o ensino-aprendizado de Sistemas Operacionais. Dentre os existentes, podemos destacar Anido (2000) e Maziero (2002).

Com base nesta proposta pedagógica, o artigo apresenta um framework construtivista para ser avaliado e utilizado no ensino-aprendizado de Sistemas Operacionais. Primeiramente, são apresentadas algumas iniciativas realizadas no ensino da disciplina como também uma visão geral do funcionamento do simulador gráfico SOsim. Conceitos da teoria construtivista são abordados, assim como possibilidades de aplicação no ensino da disciplina. Finalmente, são descritos resultados da experiência realizada no curso de Bacharelado em Sistemas de Informação da PUC-Rio.

## **2. O Ensino de Sistemas Operacionais**

Um curso de Sistemas Operacionais elaborado somente com aulas teóricas não possibilita que muitos dos conceitos sejam devidamente compreendidos e assimilados pela maioria dos alunos. É fundamental que parte da carga horária do curso seja voltada para aulas de laboratório e exercícios práticos. Neste item serão apresentadas e discutidas as práticas em laboratórios mais utilizadas na maioria dos cursos de graduação: pequenos projetos práticos, modificações no código do sistema operacional e uso de simuladores.

### **2.1. Pequenos Projetos Práticos**

Diversos pesquisadores como Ramakrishnan and Lancaster (1993), Pérez-Dávila (1995), Fekete and Greening (1996), Wagner and Ressler (1997) e Downey (1999) propõem “closed labs”, utilizando sistemas reais, na maioria dos casos, alguma versão do sistema operacional Unix. Esses trabalhos propõem laboratórios supervisionados que possibilitam aos alunos terem contato com a linguagem de comandos (shell) e o desenvolvimento de programas, utilizando system calls documentadas do sistema. Os programas desenvolvidos podem envolver algum utilitário do próprio sistema ou algoritmos clássicos, como os de comunicação e sincronização entre processos. A

maioria dos laboratórios propostos nesse grupo exigem dos alunos bons conhecimentos de programação em C, Pascal ou Java.

## **2.2. Modificação no Código do Sistema Operacional**

Os trabalhos envolvendo alteração do código fonte de um sistema operacional podem utilizar dois tipos de ambientes: sistemas reais de código aberto e sistemas educacionais. Os sistemas operacionais de código aberto, como Linux, FreeBSD e Tropix [Tropix 2003], oferecem o fonte para ser analisado e modificado. Como os sistemas abertos são desenvolvidos para oferecer alto desempenho, seus códigos são bastante difíceis de serem compreendidos por iniciantes. Além de complexo, o fonte de um sistema aberto é formado por milhares de linhas de código. Outro fator dificultador é a falta de uma documentação voltada para a utilização desses sistemas como ferramenta de ensino.

Sistemas operacionais educacionais, como Minix [Tanenbaum and Woodhull 1997] e Xinu [Comer 1984], são versões simplificadas de sistemas reais de código aberto, desenvolvido com o objetivo específico de servir para o estudo prático da estrutura interna e funcionamento de um sistema operacional. A grande vantagem dos sistemas educacionais é a existência de uma documentação voltada para auxiliar professores e alunos na tarefa de compreender o código fonte do sistema.

Independente do sistema operacional utilizado, laboratórios envolvendo a análise e alteração do código fonte do sistema oferecem uma série de dificuldades e restrições. Inicialmente, há a necessidade de professores e alunos terem excelentes conhecimentos de arquitetura de computadores, Unix e programação C/C++. O tempo necessário para instalar, alterar e depurar o sistema pode tornar o processo proibitivo. Além disso, muitas instituições não têm os recursos materiais necessários para implementar os laboratórios. Como consequência, a grande maioria dos cursos de sistemas operacionais não tem como atender estes pré-requisitos e, mesmo que tivesse, a carga horária disponível para a disciplina poderia não ser suficiente.

## **2.3. Uso de Simuladores**

Simuladores envolvem a criação de modelos dinâmicos e simplificados do mundo real. O potencial educacional deste tipo de ferramenta é muito superior ao dos programas tradicionais. Na área da ciência da computação existem simuladores que auxiliam no ensino de várias disciplinas, como redes de computadores, técnicas de programação, arquitetura de computadores e sistemas operacionais. Alguns exemplos de simuladores que podem ser citados são BASI [Bynum and Camp 1999], NACHOS [Anderson, Christopher and Procter 1999], OSP [Kifer and Smolka 1991] e RCOS.java [Chernich, Jamieson and Jones 1996, Jones and Newman 2001].

Dentre os diferentes tipos de simuladores, cada um possui suas características próprias, vantagens e desvantagens. Em geral, os simuladores, apesar de mais simples que os sistemas reais, também oferecem dificuldades em utilizá-los. A maioria dos simuladores apresenta uma elevada curva de aprendizado, exigindo também bons conhecimentos de Unix e programação.

### 3. Simulador SOsim

O SOsim é um simulador com recursos visuais que tem como principal objetivo apresentar os conceitos e técnicas encontrados nos Sistemas Operacionais multiprogramáveis modernos [Machado and Maia 2002]. Alguns dos algoritmos implementados podem ser encontrados em sistemas operacionais comerciais, como no HP OpenVMS e Microsoft Windows NT/2000/XP. O simulador emula os principais subsistemas de um sistema operacional multiprogramável, como gerência de processos, escalonamento e memória virtual por paginação.

O simulador SOsim foi desenvolvido com o propósito de ser uma ferramenta de apoio para o ensino de Sistemas Operacionais, permitindo aos professores apresentarem as técnicas e conceitos de forma clara, dinâmica e animada, melhorando a comunicação com os alunos e ampliando, assim, o poder de compreensão e entendimento [Maia 2001]. O software foi concebido para ser executado no sistema Windows da Microsoft, possuindo uma interface simples e de fácil interação e permitindo ao usuário acompanhar visualmente os diversos eventos assíncronos que ocorrem em um sistema operacional (Figura 1).

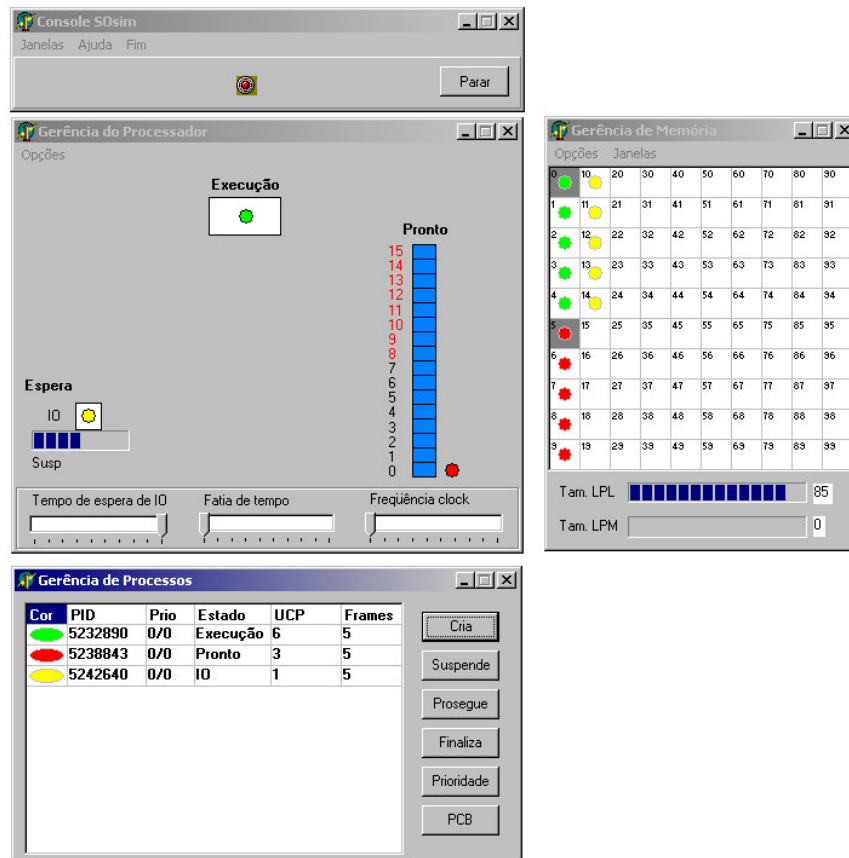


Figura 1. Simulador SOsim versão 1.2

O simulador encontra-se disponível para download gratuito na Internet nas versões em português e inglês [Maia 2004], porém o software foi desenvolvido de forma a permitir sua fácil tradução para outros idiomas. Além disto, há um grupo de usuários que está utilizando o software como ferramenta de apoio no ensino de Sistemas Operacionais, contendo, inclusive, com usuários de outros países.

#### **4. Construtivismo**

O modelo pedagógico clássico utilizado na maioria dos cursos do ensino fundamental, médio e superior no Brasil ainda possui como base o modelo instrucionista. Esse modelo é fundamentado em seqüências instrucionais com o objetivo de transferir a maior quantidade possível de informação entre um mestre ativo e um aprendiz passivo. O instrucionismo não busca um ensino individualizado para cada aluno, tendendo a ser homogêneo e padronizado.

Uma forma de transpor as limitações impostas pelo modelo instrucionista é a adoção de conceitos da teoria construtivista nos modelos pedagógicos utilizados. Neste modelo, o professor tem a função de ser um facilitador no processo de aprendizagem e não apenas um transmissor do conhecimento. O foco principal passa ser o aluno e não mais o professor. O docente tem como principal função motivar o aluno no espírito crítico-investigativo, além de orientá-lo em trabalhos cooperativos com os demais colegas na busca da construção do conhecimento.

##### **4.1. A Teoria Construtivista**

A primeira teoria construtivista foi elaborada por Jean Piaget a partir dos seus estudos iniciados na década de 1940. Suas pesquisas tiveram como base a observação de como as crianças adquirem conhecimento ao longo do seu amadurecimento. A partir desses experimentos, Piaget elaborou diversas teorias, sendo um dos aspectos mais difundidos na sua obra a descrição dos estágios de desenvolvimento cognitivo de crianças. As extensas pesquisas realizadas possibilitaram a Piaget estabelecer uma metodologia de análise que fundamentou as bases para a sua teoria de aprendizagem conhecida como Epistemologia Genética [Piaget 2002].

Piaget propõe um modelo híbrido fundamentado nos mecanismos cognitivos da espécie (sujeito epistêmico) e dos indivíduos (sujeito psicológico), de forma que o conhecimento não seja exclusivamente inerente ao próprio indivíduo e tampouco apenas desenvolvido com base no condicionamento. Na teoria construtivista, o conhecimento é assimilado por uma estruturação progressiva da experiência, evoluindo através de um processo interativo de construção. De acordo com suas teorias, o conhecimento, em qualquer nível, é gerado através de uma interação radical do sujeito com seu meio, a partir de estruturas previamente existentes no indivíduo [Ruthschilling *et. al* 1998].

O desenvolvimento do conhecimento construtivista de Piaget tem base nos mecanismos conhecidos como *assimilação* e *acomodação* que fazem parte de um processo chamado *equilíbrio*. A partir de um estágio de maturação biológica, somado à interação com os objetos e à transmissão de informações recebidas do meio, o sujeito constrói, através de associação de idéias, uma estrutura ampla de conhecimento. Caso a estrutura construída não seja consistente com as experiências vivenciadas, um erro construtivo é caracterizado, fazendo com que o sujeito reaja contrariamente à

assimilação. Neste caso, o sujeito deve reconstruir suas hipóteses até que a mesma seja passível de uma assimilação completa. Este é o mecanismo conhecido como acomodação, onde “o sujeito age no sentido de transforma-se em função das resistências impostas pelo objeto” [Maziero 2002].

O erro construtivo proporciona um desequilíbrio, possibilitando uma nova ação intelectual na busca de um novo equilíbrio. Esse processo dinâmico de construção do conhecimento com base no erro é, segundo Piaget, uma etapa necessária no desenvolvimento cognitivo. Segundo [Ruthschilling *et. al* 1998], “é imprescindível que se compreenda que sem uma atitude do objeto que perturbe as estruturas do sujeito, este não tentará acomodar-se à situação, criando uma futura assimilação do objeto, dando origem às sucessivas adaptações do sujeito ao meio, com o constante desenvolvimento de seu cognitivismo”.

#### **4.2. O Modelo Pedagógico Construtivista**

Em modelos pedagógicos desenvolvidos com base na teoria construtivista, o estudante deve construir seu próprio conhecimento e não absorvê-lo passivamente dentro da sala de aula ou através da leitura de livros-texto. Esta forma de aprendizado exige que o aluno não só descubra fatos mas que, a partir destes, construa modelos mentais viáveis que possibilitam a construção do conhecimento. Nesse método, o docente tem como função orientar os estudantes na busca desse objetivo, porém deve levar em consideração as estruturas cognitivas individuais de cada um, o que torna o método pedagogicamente mais complexo que o modelo clássico instrucionista.

Finemmann e Bootz (1995) destacam que, na teoria construtivista, os processos colaborativos e de apoio à negociação social do significado são altamente relevantes visto que cada estudante possui uma perspectiva única. O diálogo proporciona ao aprendiz expor-se a múltiplas perspectivas do ambiente, aprofundando seu entendimento através da interação com os demais colegas. Maziero (2002) afirma que a interação sujeito-objeto é base para a formação do conhecimento, pois este não está no sujeito nem no objeto, mas na ação simultânea do sujeito e do objeto um sobre o outro.

O modelo construtivista precogniza o grande benefício gerado quando alunos participam de atividades que permitam a construção ativa do seu próprio domínio do conhecimento. Para isso, é essencial que o professor seja capacitado nos fundamentos propostos por Piaget, além da criação de ambientes de ensino adequados a modelos pedagógicos construtivistas. Um dos fatores mais importantes na construção deste tipo de ambiente é a conscientização no professor da importância do educador-educando, passando todos os processos de aprendizagem por uma interação muito forte entre sujeito e objeto.

Brooks e Brooks (1993) comparam características presentes entre as salas de aula tradicionais e as construtivistas. Na Tabela 1, as principais diferenças são apresentadas.

**Tabela 1. Comparação entre salas de aula tradicionais e construtivistas**

<b>Sala de Aula Tradicional</b>	<b>Sala de Aula Construtivista</b>
Estudantes fundamentalmente trabalham sozinhos.	Estudantes fundamentalmente trabalham em grupos.

O acompanhamento rigoroso do currículo pré-estabelecido é altamente valorizado.	Busca pelas questões levantadas pelos alunos é altamente valorizada.
As atividades curriculares baseiam-se fundamentalmente em livros-texto e de exercícios.	As atividades baseiam-se em fontes primárias de dados e materiais manipuláveis.
Avaliação da aprendizagem é vista como separada do ensino e ocorre, quase que totalmente, através de testes.	Avaliação da aprendizagem está interligada ao ensino e ocorre através da observação do professor sobre o trabalho dos estudantes.

## 5. Um *Framework* Construtivista para Aprendizado de Sistemas Operacionais

Na maioria das universidades brasileiras, o modelo dominante de ensino de Sistemas Operacionais está relacionado à teoria instrucionista. O curso é fundamentado na apresentação de conceitos pelo professor, não possibilitando que os estudantes possam construir estruturas de conhecimento mentais com base no aprendizado. Esse modelo pode tornar as aulas de Sistemas Operacionais extremamente abstratas e desinteressantes. As teorias construtivistas surgem como uma opção para o desenvolvimento de propostas pedagógicas que possam obter melhores resultados que os encontrados em modelos instrucionistas.

Esse trabalho propõe um *framework* construtivista para o aprendizado de Sistemas Operacionais, a partir do qual modelos pedagógicos podem ser elaborados para a disciplina em cursos de graduação. A seguir são apresentadas as principais diretrizes para a elaboração de um modelo construtivista:

- O ensino deve ser individualizado repetindo-se o ritmo de aprendizagem de cada aluno;
- A interação aluno-professor deve ser intensa na busca por questões práticas e interessantes;
- Trabalhos em grupos deverão ser propostos para a busca de um aprendizado cooperativo;
- Um simulador de sistemas operacionais deve ser utilizado pelo professor em conjunto com as aulas teóricas, ilustrando os conceitos complexos presentes na disciplina;
- Um simulador de sistemas operacionais deve ser utilizado por alunos em trabalhos na sala de aula e em casa para simular situações de difícil reprodução em um sistema real.

Nessa proposta, a sala de aula deve possibilitar ao aluno a oportunidade de construir, com base no seu conhecimento prévio e no seu ritmo de assimilação, os novos conhecimentos propostos pela ementa do curso. É nesse contexto que o uso de um simulador de Sistemas Operacionais possibilita uma interação rica entre aluno-ferramenta, de maneira que o estudante possa confrontar-se com problemas que simulam o real. As facilidades para desenvolver hipóteses, testá-las, criar propostas de solução alternativas e debatê-las entre colegas ou com o professor, fazem do simulador uma ferramenta essencial no processo de aprendizagem. Como uma ferramenta construtivista, o simulador enfatiza a construção do conhecimento, pois proporciona

múltiplas representações da realidade, permitindo que o aluno teste suas próprias hipóteses e aprenda com seus acertos e, principalmente, com os erros obtidos. Dessa forma, o software favorece o pensamento reflexivo em função da experiência com o ambiente de simulação. Além disso, permite ao aluno o controle do experimento, garantindo uma evolução natural da complexidade das simulações. O simulador permite que, dado um problema, o aluno busque ativamente uma solução, exercitando, assim, a identificação, definição e solução de problemas.

## **6. Um Modelo Pedagógico Construtivista Experimental**

Um modelo pedagógico experimental com base no framework construtivista foi elaborado para utilização na disciplina de Sistemas Operacionais do curso de Bacharelado em Sistemas de Informação da PUC-Rio. Esse modelo utiliza o simulador SOsim em aulas de laboratório e inclui a realização de trabalhos em grupos nas aulas práticas e para serem realizados em casa.

O simulador SOsim foi adotado por ser uma ferramenta de simples aprendizado e que consegue ilustrar de forma fácil e intuitiva conceitos teóricos, permitindo ao aluno construir o seu modelo mental de conhecimento. Atualmente, três aulas de laboratório são reservadas no programa do curso para atividades em grupo com o uso do simulador. A primeira aula explora os conceitos relacionados à gerência de processos, a segunda trabalha as diversas políticas de escalonamento e última aula aborda o mecanismo de memória virtual com paginação.

As aulas de laboratório são realizadas logo após a parte teórica ser apresentada em sala de aula. Os alunos são levados para um laboratório e divididos em grupos de dois. Os grupos recebem uma folha de atividades contendo exercícios práticos, propostas de simulações a serem analisadas e algumas questões teóricas que devem ser respondidas com auxílio da ferramenta. Ao longo da aula, os alunos e o professor discutem os resultados que vão sendo obtidos em suas simulações e trocam experiências.

Durante o ano de 2003 foi realizada uma pesquisa quantitativa e qualitativa entre os alunos após a realização de cada laboratório. O objetivo era avaliar os resultados do uso do simulador como instrumento válido para uma metodologia pedagógica com proposta construtivista. A pesquisa continha sete questões e foi aplicada para uma amostra de trinta alunos. Com exceção das questões seis e sete, que poderiam ser respondidas livremente, todas as demais apresentaram o mesmo tipo de estrutura com escala de respostas com cinco valores: “Discordo Totalmente”, “Discordo em Parte”, “Não Concordo, Nem Discordo”, “Concordo em Parte” e “Concordo Totalmente”. As perguntas que fizeram parte da pesquisa foram:

1. O uso do simulador torna o aprendizado dos conceitos teóricos mais agradável.
2. O uso do simulador ajuda a despertar o interesse pelo assunto.
3. O uso do simulador é fácil e sua interface é clara.
4. O uso do simulador auxilia na compreensão e na assimilação dos conceitos teóricos.



5. O uso do simulador é interessante para simular situações reais em um sistema operacional.
6. Quais os aspectos do simulador você acha mais interessante para o aprendizado de sistemas operacionais?
7. Qual a sua opinião sobre o uso do simulador?

O resultado das questões objetivas pode ser observado na Tabela 2. Em resumo, a pesquisa mostra que a maioria dos alunos achou que o aprendizado foi mais agradável, despertou o interesse sobre o assunto, auxiliou na compreensão dos conceitos e permitiu simular situações reais.

**Tabela 2. Resultados da Pesquisa**

Questão	Discordo Totalmente	Discordo em parte	Nem concordo nem discordo	Concordo em parte	Concordo totalmente
Q1				15,79%	84,21%
Q2			15,79%	42,11%	42,11%
Q3		10,53%	21,05%	52,63%	15,79%
Q4			5,26%	15,79%	78,95%
Q5			5,26%	52,63%	42,11%

Na questão seis, os alunos responderam, na sua maioria, que o software ajudou-os a visualizar conceitos e problemas referentes a sistemas operacionais, além de aproximar a teoria à prática. Na pergunta sete, a maioria dos alunos apoiou e elogiou a iniciativa de utilização do software. Alguns alunos pediram um número maior de aulas de laboratório e outros sugeriram melhorias no software. A seguir algumas respostas coletadas na pesquisa:

- “O simulador ajuda o aluno a sair um pouco da teoria.”
- “A visualização dos conceitos dados em sala de aula.”
- “Muito bom. Aprendi muito.”
- “Poderiam haver mais aulas de laboratório, pois o simulador ajuda muito a assimilar os conceitos.”
- “Acho excelente, porém acho que poderiam ser inseridas outras ferramentas.”
- “A idéia é boa, mas o software tem que ser ainda muito aprimorado.”

## 7. Conclusões

A adoção de um modelo pedagógico construtivista permite oferecer excelentes perspectivas para a melhoria no ensino-aprendizado de Sistemas Operacionais. A partir do modelo experimental que vem sendo utilizado na PUC-Rio, as dificuldades encontradas no ensino de um assunto não-linear e na associação da teoria e prática, puderam ser atenuadas, conforme indicam os resultados da pesquisa realizada. Além disto, espera-se um grande avanço qualitativo no processo de aprendizado, que deverá ser percebido, principalmente, em outros tipos de cursos, onde a carga horária estabelecida para a disciplina é menor ou nos cursos que tenham alunos que não possuam os pré-requisitos necessários.

Na experiência realizada, o uso do simulador SOsim mostrou-se importante no auxílio da construção do saber, pois possibilitou a maior interação do aprendiz com o objeto de estudo. Uma grande vantagem pedagógica em utilizar tal ferramenta é a construção de um ambiente híbrido de ensino-aprendizado, onde aulas expositivas tradicionais e simulações podem ser combinadas. Dessa forma, o colaboracionismo é aplicado em sala de aula e o pensamento construtivista pode ser introduzido como apoio à formação do conhecimento, permitindo a experimentação das teorias apresentadas. A utilização do simulador pode reduzir o tempo necessário para as apresentações conceituais, permitindo ampliar a parte prática da disciplina e criar novos laboratórios.

É importante que alguns problemas sejam destacados no uso de um modelo construtivista. Uma crítica usual às teorias de Piaget é justamente a falta de uma prática pedagógica clara e explícita, uma vez que não é isto a que ela se propõe. A razão está no princípio teórico de um modelo que é, por princípio, epistemológico e, não, pedagógico. Outro grande problema relevante é a falta de preparo pedagógico da maioria dos professores. O docente construtivista deve estar familiarizado com os princípios propostos por Piaget e ser conhecedor do modelo pedagógico proposto.

A continuidade deste trabalho visa ampliar e aprimorar o modelo construtivista proposto, de forma a estruturar uma prática pedagógica explícita para o ensino de Sistemas Operacionais. Com isso, é esperado estabelecer um método pedagógico que possa ser utilizado por docentes que não tenham o conhecimento prévio das teorias de Piaget. Além disto, novas potencialidades do simulador podem ser exploradas, de forma a ampliar os benefícios da prática construtivista. Novas análises comparativas entre os métodos construtivistas e tradicionais para o ensino de Sistemas Operacionais devem ser realizadas com detalhes, além de avaliações e pesquisas com a finalidade de aferir a eficácia do método.

## **8. Referências**

- Anderson, T. E., Christopher, W. A. and Procter, S. J. (1999). "The Nachos instructional operating system". Disponível via Internet em <http://www.cs.washington.edu/homes/tom/nachos/>.
- Anido, R. (2000). "Uma proposta de plano pedagógico para a matéria de Sistemas Operacionais". Anais do II Curso de Qualidade sobre Educação em Computação, XX Congresso da SBC.
- Ben-Ari, M. (1998). "Constructivism in computer science education". Proceedings of the 29<sup>th</sup> ACM SIGCSE.
- Brooks, J. G. and Brooks, M. G. (1993). "Structuring learning around primary concepts: The quest for essence. In search of understanding: The case for constructivist classrooms". Alexandria, VA: Association for Supervision and Curriculum Development.
- Bynum, B. and Camp, T. (1999). "After you, Alfonse: a mutual exclusion toolkit - an introduction to BASI". Disponível via Internet em [http://www.mines.edu/fs\\_home/tcamp/baci/](http://www.mines.edu/fs_home/tcamp/baci/).

- Chernich, R., Jamieson, B. and Jones, D. (1996). "RCOS: Yet another teaching operating system". Proceedings of the 1<sup>st</sup> Australian Conference on Computer Science Education.
- Comer, D. Operating System Design – The XINU Approach. Prentice-Hall, 1984.
- DC-MEC. "Diretivas Curriculares-Ministério da Educação e Cultura". Disponível via Internet em [http://www.mec.gov.br/sesu/ftp/curdiretriz/computacao/co\\_diretriz.rtf](http://www.mec.gov.br/sesu/ftp/curdiretriz/computacao/co_diretriz.rtf).
- Downey, A. B. (1999). "Teaching experimental design in an operating systems class". Proceedings of the 30<sup>th</sup> ACM SIGCSE.
- Fekete, A. and Greening, A. (1996). "Designing closed laboratories for a computer science course". Proceedings of the 27<sup>th</sup> ACM SIGCSE.
- Finemman, E. and Bootz, S. An Introduction to constructivism in Instructional Design. Technology and Teacher Education Annual University of Texas, 1995.
- Jones, D. and Newman, A. (2001). "RCOS.java: a simulated operating system with animations". Proceedings of the Computer-Based Learning in Science Conference, Brno, Rep. Tcheca.
- Jones, D. and Newman, A. (2002) "A Constructivist-based tool for operating systems education". Disponível na Internet em [http://cq-pan.cqu.edu.au/david-jones/Publications/Papers\\_and\\_Books/RCOS.java\\_2002/](http://cq-pan.cqu.edu.au/david-jones/Publications/Papers_and_Books/RCOS.java_2002/).
- Kifer, M. and Smolka, S. OSP: An Environment for Operating Systems (Instructor Version). Addison-Wesley, 1991.
- Machado, F. B. and Maia, L. P. Arquitetura de Sistemas Operacionais. 3<sup>o</sup> ed., LTC, 2002.
- Maia, L. P. (2001). "SOsim: Simulador para o Ensino de Sistemas Operacionais". Tese de Mestrado, Núcleo de Computação Eletrônica da Universidade Federal do Rio de Janeiro (NCE/UFRJ), Março.
- Maia, L. P. (2004). "Página do SOsim". Disponível na Internet em <http://www.training.com.br/sosim>.
- Maziero, C. A. (2002). "Reflexões sobre o ensino prático de Sistemas Operacionais". Anais do X Workshop sobre Educação em Computação (WEI2002), XXII Congresso da SBC.
- Pérez-Dávila, A. (1995). "OS bridge between academia and reality," Proceedings of the 26<sup>th</sup> ACM SIGCSE.
- Piaget, Jean. Epistemologia Genética. Ed. Martins Fontes, 2002.
- Ramakrishnan, S. and Lancaster, A. M. (1993). "Operating system projects: linking theory, practice, and use". Proceedings of the 24<sup>th</sup> ACM SIGCSE.
- Ruthschilling, Evelise A. *et al* (1998). "A evolução dos ambientes de aprendizagem construtivistas". Disponível na Internet em <http://penta.ufrgs.br/~luis/Ativ1/AmbApC.html>.
- Tanenbaum, A. S. and Woodhull, A. S. Operating System: Design and Implementation, 2<sup>o</sup> ed., Prentice-Hall, 1997.

Tropix (2003). "TROPIX Distribuição e Instalação". Disponível na Internet em <http://www.tropix.nce.ufrj.br>.

Wagner, T. D. and Ressler, E. K. (1997). "A practical approach to reinforcing concepts in introductory operating systems". Proceedings of the 28<sup>th</sup> ACM SIGCSE.