



COBENGE 2005

XXXIII - Congresso Brasileiro de Ensino de Engenharia

"Promovendo e valorizando a engenharia em um cenário de constantes mudanças"

12 a 15 de setembro - Campina Grande - Pb

Promoção/Organização: ABENGE/UFPE

ENSINO DOS CONCEITOS BÁSICOS DE COMPUTAÇÃO AOS INGRESSANTES DE UM CURSO DE ENGENHARIA DE COMPUTAÇÃO – RESULTADOS ALCANÇADOS

Ricardo Luis de Azevedo da Rocha – luis.rocha@poli.usp.br

Escola Politécnica da Universidade de São Paulo, Departamento de Engenharia de Computação e Sistemas Digitais – PCS

Av. Prof. Luciano Gualberto, trav.3, nº.158

05508-900 – São Paulo – SP

Sidney de Castro – sidney.castro@fsa.br

Faculdade de Engenharia da Fundação Santo André, Curso de Engenharia da Computação

Av. Príncipe de Gales, 821

09060-650 – Santo André – SP

Dirceu da Silva – dirceu@unicamp.br

Faculdade de Educação da Unicamp, Departamento de Metodologia do Ensino – Educação, Ciência e Tecnologia

Rua Bertrand Russel nº 801

13083-865 – Campinas – SP

***Resumo:** O uso de um modelo formal de computação (a máquina de Turing) para ensinar conceitos básicos aos alunos ingressantes foi proposto em ROCHA et al (2004), a partir de uma experiência em andamento em uma escola de engenharia no grande ABC paulista. Um aspecto bastante interessante desta proposta é a equalização do conhecimento que os alunos têm com respeito ao conteúdo da disciplina ao se introduzir a máquina de Turing como ponto de partida. /Todos os alunos estão em igualdade de condições, já que nenhum deles conhece a máquina de Turing. Cada aluno pode ter evolução diferenciada na disciplina, evidenciando seus talentos, que podem ser aproveitados pelo professor. Comparamos o desempenho dos alunos em uma disciplina do curso, que não sofreu alteração desde antes da introdução do modelo proposto. Os resultados qualitativos e quantitativos obtidos permitem concluir que houve melhoria de desempenho com a introdução do modelo. Pequenas mudanças são propostas para aumentar a sinergia entre as disciplinas do curso.*

***Palavras-chave:** Metodologia de Ensino de Conceitos Básicos em Computação, Algoritmos, Máquina de Turing*

1. INTRODUÇÃO

O uso de um modelo formal de computação (a máquina de Turing) para ensinar conceitos básicos aos alunos ingressantes foi proposto em ROCHA et al (2004), a partir de uma experiência em andamento em uma escola de engenharia no grande ABC paulista. Conforme exposto no artigo citado, há uma enorme dificuldade por parte do corpo discente em perceber a necessidade de compreensão dos conceitos básicos, aliada a uma forte tendência imediatista, típica desta geração, a geração “Nintendo” conforme GUZDIAL e SOLOWAY (2002).

Na proposta formulada os conceitos de algoritmos e estruturas de controle, máquinas de Turing, linguagens de programação e estruturas de dados, máquinas universais de computação (computadores universais) e traduções são desenvolvidos ao longo de um semestre em uma disciplina denominada “Introdução à Informática”. O ponto de partida para trabalhar tantos conceitos importantes, ainda que de forma introdutória, é a utilização do modelo de máquina de Turing em problemas, projetos e desafios, de maneira a motivar o aluno a descobrir respostas, soluções.

A cada nova situação-problema o aluno apreende novos conceitos, que depois são formalizados pelo professor. Desta maneira, quando o aluno se depara com um problema, que é uma generalização de outro já estudado, percebe a necessidade de organizar melhor as estruturas de controle de uma máquina de Turing. Com isso, percebe que se há alguma estruturação física, uma arquitetura previamente definida na máquina, a generalização se faz de forma mais direta. Assim sendo, os conceitos são introduzidos por etapas, conforme o aluno percebe a necessidade de alguma formulação da qual não tem conhecimento.

Um aspecto bastante interessante desta proposta é a equalização do conhecimento que os alunos têm com respeito ao conteúdo da disciplina, visto que muitos alunos que se matriculam em um curso de computação não têm conhecimento prévio na área, e que os demais (que possuem algum tipo de conhecimento) conhecem pouco em termos conceituais e detêm conhecimento tecnológico. Portanto, ao se introduzir a máquina de Turing como ponto de partida, todos estão em igualdade de condições, isto é, no marco inicial. O professor tem, então, condições de trabalhar o conteúdo da disciplina sem se preocupar em demasia com as diferenças entre o conhecimento prévio dos alunos.

Assim, cada aluno pode ter evolução diferenciada na disciplina, e seus conhecimentos prévios aliados a seus talentos intrínsecos o permitem trilhar caminhos diversos dentro da própria disciplina. A esse respeito, na seção 5 ilustramos com o caso de um grupo pequeno de alunos que possuem talentos semelhantes e se uniram para desenvolver um complemento ao simulador utilizado, o VTuring¹.

A proposta desenvolvida para a disciplina de “Introdução à Informática” leva em consideração as dificuldades enfrentadas pelos alunos e tem como objetivo melhorar a compreensão global dos conceitos básicos de computação. Por esta razão, esperamos um desempenho melhor nas disciplinas subseqüentes, como, por exemplo, na disciplina de “Algoritmos Computacionais”, na qual são formalizados adequadamente os processos de construção e descrição para soluções de problemas computacionalmente possíveis.

Assim sendo, avaliamos os resultados obtidos na disciplina de “Algoritmos Computacionais” das turmas ingressantes de 2003, nas quais o modelo proposto ainda não havia sido implantado, e das turmas ingressantes de 2004, nas quais o modelo foi implantado, buscando identificar a influência do modelo proposto para a disciplina que a antecede, “Introdução à Informática”.

¹ Pode ser encontrado em www.cheransoft.com/vturing

Portanto, na seção 2 apresentamos como resultado a avaliação comparativa destes dois grupos na disciplina de “Algoritmos Computacionais”. Os resultados permitem concluir que os conceitos foram apreendidos mais profundamente pela turma que se beneficiou da proposta.

Na última seção apresentamos as mudanças introduzidas na disciplina, a partir deste ano, que visam melhorar a forma de trabalhar alguns aspectos computacionais importantes e, ao mesmo tempo, permitir maior participação do corpo discente na produção de material didático e conteúdo técnico.

2. AVALIAÇÃO DO DESEMPENHO DOS ALUNOS

Os resultados preliminares a respeito do desempenho dos alunos na disciplina de “Algoritmos Computacionais” tanto do ano de 2003 quanto de 2004, apontam para uma dispersão menor da turma de 2004. Os resultados são:

Turma de 2003, matutino, (média = 7,088, variância = 2,910);

Turma de 2003, noturno, (média = 7,0662, variância = 2,7597);

Turma de 2004, matutino, (média = 6,193, variância = 1,808);

Turma de 2004, noturno, (média = 7,1869, variância = 1,8788);

Observamos que a variância das turmas de 2004 é menor do que as das turmas de 2003, com valores para as médias semelhantes, embora a média da turma do matutino de 2004 também seja menor do que a da turma de 2003. Este resultado evidencia que os alunos têm desempenho médio semelhante, afinal sua origem não é distinta, mas houve alguma diferença no processo de ensino-aprendizagem que tornou a dispersão menor. Portanto, os alunos de 2004 estão em nível de aprendizado, conhecimento e desenvoltura mais próximos do que no ano de 2003.

A dispersão entre os alunos do matutino e do noturno não é diferente para alunos do mesmo ano, o que corrobora com a afirmação de que o resultado medido tem sua origem no processo de ensino-aprendizagem utilizado (variâncias 2003: matutino – 2,910, noturno – 2,7597; variâncias 2004: matutino – 1,808, noturno – 1,8788).

O teste F aplicado a uma distribuição de dados permite obter uma comprovação de que as diferenças entre as variâncias não sejam significativas, ou seja, o seu resultado indica a probabilidade de que as distribuições não tenham variâncias diferentes. Aplicando o teste F aos nossos dados obtivemos (TesteF noturno = 0,128304, 12,83%; TesteF matutino = 0,110236, 11,02%). Estes valores são pequenos, se comparados aos valores obtidos para a comparação entre os períodos matutino e noturno do mesmo ano (2003: 0,825547, 82,55%; 2004: 0,911766, 91,18%).

Pode-se concluir que houve melhora na variabilidade dos dados, indicando menor dispersão. Estes resultados foram obtidos através da introdução de uma nova variável, o modelo proposto em ROCHA et al (2004), já que as outras variáveis não foram alteradas. Em anexo encaminhamos os dados utilizados para a comparação efetuada.

3. ASPECTOS QUALITATIVOS DO PONTO DE VISTA PEDAGÓGICO

Os alunos ingressantes de 2004 que permaneceram matriculados ao longo desse mesmo ano apresentaram desempenho superior aos alunos que ingressaram em 2003 e que permaneceram matriculados naquele ano. Esta constatação decorre de avaliação qualitativa efetuada junto aos professores que ministraram disciplinas a essas turmas no segundo semestre de 2003 e 2004 (foram os mesmos professores).

Dentre os principais aspectos mencionados por todos os professores está a maior facilidade de compreensão de modelos formais e o aumento da capacidade de resolver problemas, sempre comparando com a turma de 2003. Nem todas as disciplinas mantiveram a proposta e o conteúdo programático planejado, devido à reestruturação do primeiro ano do curso. Por conta disso, para a maioria das disciplinas somente é possível efetuar uma avaliação qualitativa das diferenças.

A disciplina de “Algoritmos Computacionais” manteve a estrutura original, permitindo com isso que efetuássemos também uma avaliação quantitativa das diferenças entre os alunos. Por conta disso, esta disciplina foi escolhida como parâmetro comparativo entre os alunos ingressantes de 2003 e 2004.

Quanto aos aspectos pedagógicos, avaliados qualitativamente pelo professor da disciplina, percebemos claramente que o professor mostra-se bastante motivado com a melhoria de desempenho dos alunos. Discorrendo sobre os diversos fatores que considera mais adequados na turma que ingressou em 2004, as seguintes considerações foram expostas:

- Melhor entendimento do conceito de programa pelos alunos, que passaram a criar blocos lógicos usando uma notação mais matemática, ou seja, um maior rigor. A arquitetura e a forma de resolver o problema na disciplina anterior “Introdução à Informática” conduzia o aluno a quebrar o todo em pedaços lógicos; e a rigidez da notação (máquina de Turing) conduzia o aluno a agir de forma mais organizada.
- Melhor avaliação da função como um ato computacional independente. O aluno passou a identificar a possibilidade de reaproveitar funções, mantendo-as desacopladas, o que propiciou melhoria na percepção do conceito de acoplamento.
- Percepção do caráter mecânico da computação. Decorrente do rigor matemático e da necessidade de planejamento estratégico exigidos na disciplina de “Introdução à Informática” como etapa para a resolução de problemas.

Um exemplo citado pelo professor, trata de um exercício de conversão de Celsius para Fahrenheit, cujo objetivo foi realizar uma função para efetuar a conversão através da fórmula matemática “ $c/5 = (f-32)/9$ ”. Segundo o professor, antes do uso deste modelo (em 2003), a resposta apresentada era invariavelmente uma repetição explícita do problema, sem identificar a necessidade de isolar a variável. Por outro lado, para os alunos ingressantes em 2004, houve naturalmente a percepção da necessidade de isolar as variáveis.

O professor considera o desempenho global dos alunos ingressantes em 2004 superior ao dos alunos ingressantes em 2003, não apenas pelo rendimento mensurável em notas, mas, principalmente, pela forma com que o processo de aprendizagem se desenrolou ao longo do semestre na disciplina. Segundo o professor, os alunos apresentaram um nível de compreensão mais próximo do esperado e, com isso, a turma mostrou-se mais homogênea.

A constatação do professor é muito semelhante à dos demais colegas, o que corrobora com uma avaliação bastante positiva do modelo proposto para a disciplina de “Introdução à Informática”. Os alunos passaram a apresentar maior capacidade de lidar com formalismo matemático, que é indispensável a um engenheiro de computação.

Um outro aspecto relevante e almejado em nossa proposta diz respeito à capacidade de compreensão do problema em todos os seus aspectos, para que seja possível resolvê-lo da maneira mais adequada. Segundo constatação qualitativa dos professores este aspecto revelou-se também fortalecido pelo maior contingente de alunos ingressantes em 2004, em relação às turmas ingressantes em 2003, que se mostrou apto a ler e a compreender os problemas propostos.

4. EVOLUÇÃO DA PROPOSTA – INGRESSANTES EM 2005

Para o ano de 2005 propusemos algumas alterações simples na disciplina, de forma a melhorar o seu encadeamento lógico com outra disciplina do segundo semestre do curso - a disciplina de “Laboratório de Programação de Computadores” - na qual é estudada e trabalhada a linguagem de programação Java.

Para que os alunos possam apreender e utilizar os conceitos de orientação a objetos, é necessário que o modelo introduzido permita o desenvolvimento do conceito de objetos, a partir do conceito de máquina de Turing. Neste sentido, a disciplina de “Introdução à Informática” passou a utilizar, no seu terço final, exercícios-problema que justifiquem a necessidade de maior abstração, de reutilização, enfim, de conceitos de orientação a objetos.

O modelo de máquina de Turing utilizado ao longo do semestre apresenta as seguintes características, conforme descrito em ROCHA et al, (2004):

- *stin*: função de entrada da máquina, com dois parâmetros o registrador e o valor do dado;
- registradores: três ao todo (representados em nosso caso por “u”, “d”, e “s”) cada registrador tem 8 células ao todo.

Usando este modelo o aluno deverá ser capaz de descrever as soluções para os problemas propostos definindo uma estratégia implementável e, posteriormente, descrevendo a sua estrutura da solução em uma linguagem simples, semelhante a uma linguagem “*assembly*” para o modelo de máquina de Turing proposto.

Em nossa proposta, a arquitetura de computador para desenvolver a solução é representada por uma máquina de Turing, na qual a fita de entrada possui indicações específicas dos registradores, com marcadores de início e fim de dados. A arquitetura é utilizada para implementar a solução, que deve ser visualizada como uma seqüência de instruções na linguagem “*assembly*” proposta. Desta maneira, o aluno integra os conceitos de hardware e software, devendo programar o hardware para executar as tarefas designadas pelas instruções escritas em “*assembly*”.

Um modelo de exercício-problema é apresentar um programa na linguagem “*assembly*” da máquina e solicitar a descrição detalhada na máquina de Turing com as instruções necessárias para realizar as tarefas do programa. Desta maneira, o aluno compreende a separação entre hardware e software como algo artificial e é capaz de construir o modelo de máquina de Turing para implementar o programa usando um conjunto de outras máquinas menores, operando em conjunto, tal como um pequeno pacote de classes.

Outro modelo de exercício-problema apresentado é oferecer somente uma descrição de um problema, sem o programa associado, e solicitar que se encontre uma solução. Em ambos os tipos de exercício, observamos que o número de alunos cujas respostas excediam em muito o mínimo esperado aumentava a cada novo exercício solicitado.

Ao final do semestre, a grande maioria dos alunos conseguiu empreender algum tipo de solução para a maioria dos problemas apresentados, cada qual a seu próprio tempo, e eles próprios comparavam as soluções apresentadas, buscando apontar o que havia de melhor nelas.

4.1. Exemplo ilustrativo – exercício-problema descrito em linguagem

O exercício a seguir está representado na notação de linguagem “*assembly*” utilizada e há uma explicação na coluna da direita (após os símbolos //) que descreve a ação a ser efetuada em

cada linha. Cada linha deve ser descrita como uma ou mais máquinas de Turing, cujo objetivo é realizar a tarefa solicitada.

Multiplifica:

```
stin(U,TEC); // Para U > 0,
multiplicando.stin(D,TEC); // Para D > 0
multiplicador.'ZerR'(); // Zerar R
RJump('VeZD()', Sai, Multiplicar); // Decisão
Multiplicar: // ref. para o retorno.
'DecD'(); // Decremento do registrador D.
'CpSU'(); // Calcula R=R+U
Jump('VeZD()', Sai, Multiplicar); // Decisão (salto condicional).
Sai:
halt
```

Aproximadamente 1/3 dos alunos presentes na sala de aula conseguiram apresentar uma solução adequada para o problema acima em tempo hábil, ou seja, em 90 minutos, sem qualquer tipo de ajuda externa. Os demais alunos levaram mais tempo, cada qual segundo sua própria característica, mas, nenhum aluno ficou sem apresentar uma solução em até um prazo de duas aulas após a exposição ao problema.

Cada aluno apresentou sua própria solução em um relatório que descreve a estratégia adotada (com a devida formalização matemática), a funcionalidade de cada máquina de Turing mais simples, a cópia do modelo gráfico gerado pelo programa VTuring e uma cópia da tabela de transição implementada. O fator mais interessante observado é a variedade de soluções apresentadas, algumas com alto grau de refinamento.

No endereço eletrônico <http://200.229.137.19/NewFAENG/computacao/IntroInfo/Turing> há a descrição dos trabalhos solicitados na disciplina de “Introdução à Informática”, material de apoio, enfim, é um sítio na Internet através do qual o aluno pode buscar mais informações, comunicar-se com o professor responsável, tornar-se mais autônomo e consciente de seu papel no processo de ensino-aprendizado.

5. PARTICIPAÇÃO DO CORPO DISCENTE NA PRODUÇÃO DE MATERIAL DIDÁTICO

A apresentação dos resultados das tarefas é feita usando-se inicialmente o modelo gráfico gerado pela ferramenta VTuring. Em um segundo passo é necessário apresentar os resultados como uma tabela de transições de estado, na qual os estados da máquina de Turing padrão subjacente devem ser identificados. Esta transformação de resultados é feita de forma manual, cabendo ao aluno gerar a transformação e escrever a tabela de transição à mão.

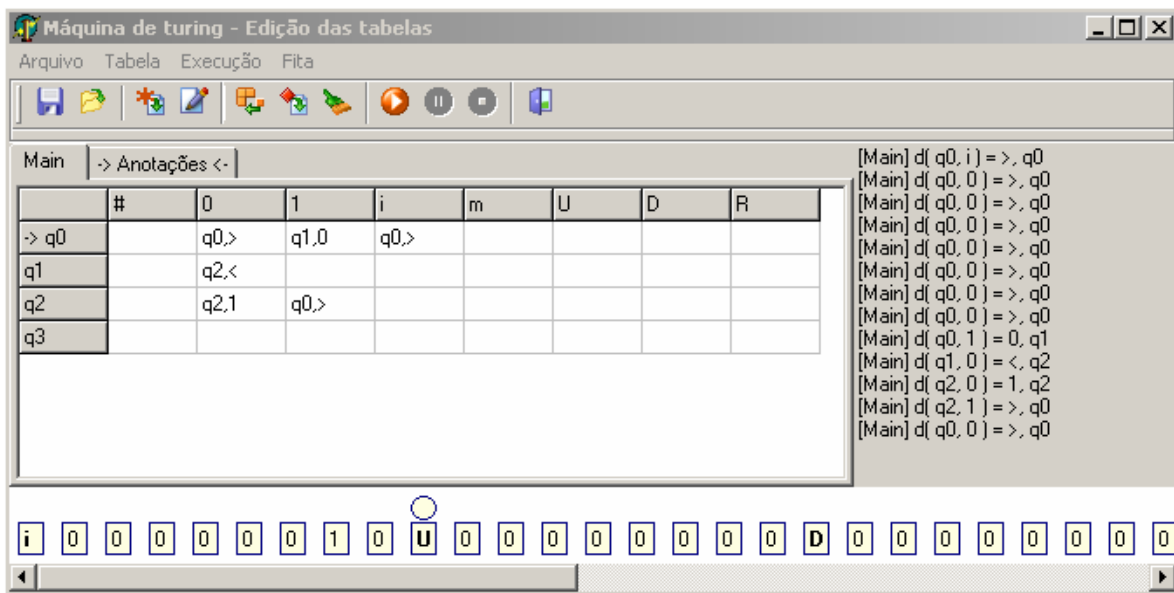
Esta tarefa é de extrema importância para demonstrar o conhecimento da estrutura interna da arquitetura, bem como para garantir que a máquina projetada efetue exatamente o que foi solicitado no exercício-problema. Entretanto, a execução manual da transformação é penosa, e não permite efetuar simulações complexas.

Por conta desta limitação, um grupo de alunos tomou para si a tarefa de codificar um programa, baseado na descrição da máquina de Turing, que oferecesse a possibilidade de simulação das alternativas de solução. Há simuladores prontos para a execução da tarefa, mas nossa intenção é aumentar a participação dos alunos, promovendo aumento na motivação e incentivando-os a buscar alternativas diferentes.

Assim sendo, os alunos foram encorajados a realizar a tarefa e o aluno líder, por ser programador Delphi, tomou para si o encargo de definir o protótipo inicial. Neste protótipo os alunos têm condições de escrever as suas soluções, testá-las, salvá-las, e, posteriormente, imprimi-las.

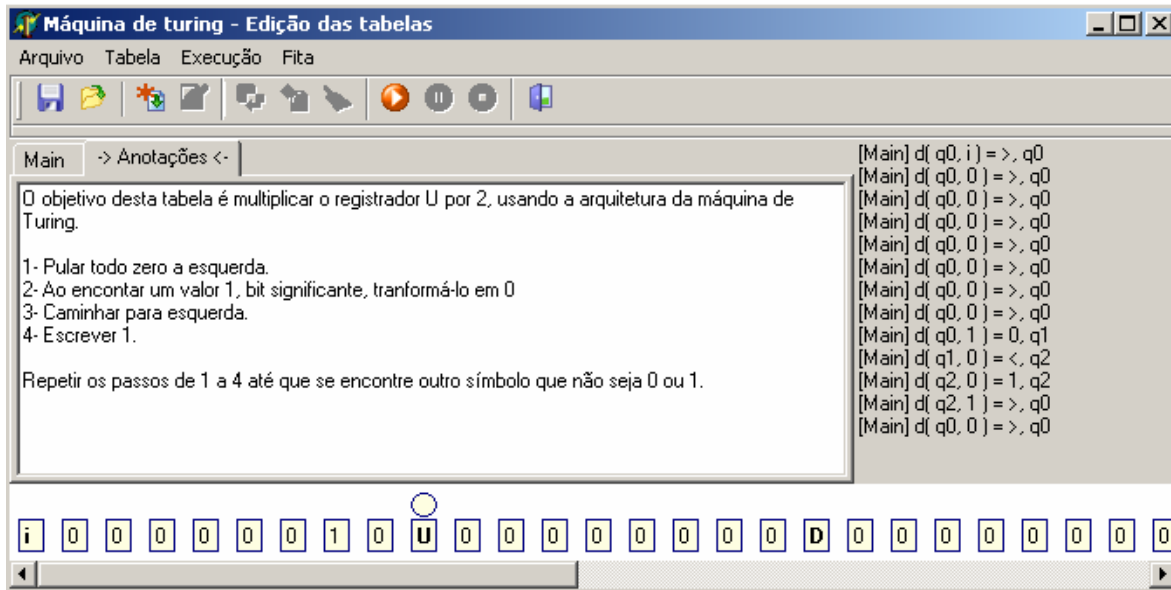
O programa também tem a função de permitir que o aluno comente e faça anotações, tornando mais clara a sua apresentação da solução. Possui uma interface de programação (que contempla a tabela de transições de estados) e uma interface de anotações (que contempla a inserção de comentários e explicações). Segue na figura 1 abaixo um exemplo de tela do programa, para construção da tabela de transições:

Figura 1 – Programa desenvolvido pelos alunos



A janela de anotações está exemplificada na figura 2 a seguir:

Figura 2 – Janela de anotações do programa desenvolvido pelos alunos



Com a ferramenta construída, os alunos puderam melhorar as suas soluções, por dispor de um simulador apropriado e construído por eles próprios. O alcance da ferramenta não pode ser medido apenas pelo uso na disciplina, mas, principalmente, pelo efeito motivador e multiplicador no corpo discente.

6. CONSIDERAÇÕES FINAIS

O nosso aluno tem a chance de testar a sua possível solução sem interferência de outros elementos condicionantes. Esta possibilidade garante que o aluno tenha uma oportunidade de expressar seus anseios mais facilmente, fazendo com que ele ganhe autonomia e possa exibir alto grau de ousadia.

Conforme mencionado em ROCHA et al (2004), o aluno passa a ter uma visão mais completa da solução e tem a possibilidade de visualizar aplicações práticas dos problemas em situações reais, já que a sua solução, bem como o problema original estão em um nível de abstração maior. Outro aspecto é a percepção dos limites de computação, não apenas pelo fato da máquina estar limitada, mas de haver problemas que sequer podem ser modelados como um processo computacional.

O aluno passa a ter condições de perceber como determinadas soluções são piores do que outras, seja por se tornarem mais lentas, ou por ocuparem mais recursos da máquina, usando ambos os simuladores (VTuring e a solução proposta por eles mesmos para a tabela de transição de estados). O efeito disso foi favorecer uma “competição” entre os alunos para o melhor modelo de solução.

A proposta formulada em ROCHA et al (2004) posta em prática resultou em uma efetiva e perceptível melhora no desempenho dos nossos alunos, tanto do ponto de vista qualitativo, quanto no ponto de vista quantitativo, embora ainda haja pequenas questões a serem resolvidas.

As melhorias propostas para 2005 estão sendo implementadas (o semestre letivo ainda não acabou), mas já há resultados, como, por exemplo, um simulador desenvolvido pelos próprios

alunos. Com isso, a participação e a integração do corpo discente foram sensivelmente maiores do que em 2004.

Além disso, trabalhamos com a integração dos conceitos de orientação a objetos, ao aumentar o grau de abstração nos exercícios-problema e na apresentação das soluções, buscando a reusabilidade de software. Assim, a transição para a linguagem de programação orientada a objetos Java deverá ser efetuada de forma menos traumática e, esperamos, de maneira mais simples do que para a turma de 2004. Estas pequenas mudanças tiveram reflexo também em outras disciplinas, como, por exemplo, em “Introdução à Engenharia da Computação”.

Agradecimentos

Agradecemos ao Prof. Heinar Augusto Weiderpass por suas contribuições e participação na análise dos dados quantitativos e qualitativos dos alunos na disciplina sob sua responsabilidade – “Algoritmos Computacionais”.

Agradecemos ainda ao aluno Bruno Martins Stuaní por sua dedicação e permissão de uso do programa desenvolvido na disciplina de “Introdução à Informática”.

REFERÊNCIAS BIBLIOGRÁFICAS

GUZDIAL, M.; SOLOWAY, E. Teaching the Nintendo Generation to Program. **Communications of the ACM**. v. 45, n. 4, p. 17-21, 2002.

ROCHA, R. L. A.; CASTRO, S.; SILVA, D. Uma Proposta de Ensino dos Conceitos Básicos de Computação aos Ingressantes de um Curso de Engenharia de Computação. In: CONGRESSO BRASILEIRO DE ENSINO DE ENGENHARIA – COBENGE’2004, 2004, Brasília. **Anais Brasília: UnB. CD-ROM.**

TEACHING BASIC CONCEPTS OF COMPUTATION TO FRESHMEN UNDERGRADUATE STUDENTS OF COMPUTATION ENGINEERING – RESULTS ACHIEVED

***Abstract:** The use of formal model of computation (the Turing machine) to teach basic concepts to freshmen undergraduate students was proposed in ROCHA et al (2004), this proposal was based on an experiment undergoing at an engineering school in the ABC (São Paulo). A particularly interesting aspect of this proposal is the equalization of the knowledge of the students, in terms of their backgrounds on the course, when the Turing machine is introduced as the starting point. All of them are equal, since none of them know anything about the Turing machine. Students may evolve differently on the subject of the course, showing their own talents, which can be exploited by the teacher. We compared the performance of the students in a course that has not been changed since before the introduction of the proposed model. The qualitative and quantitative results obtained allow us to conclude that we had an improvement on the performance with the introduction of the model. Little changes are proposed in order to increase the synergy among the courses.*

***Key-words:** Teaching Methodology of Basic Concepts of Computation, Algorithms, Turing Machine*

ANEXO – Médias de “Algoritmos para Computação”

Matutino		Noturno	
2003	2004	2003	2004
7,4	5,0	7,4	8
5,5	6,0	5,5	7,3
7,1	5,6	7,1	7,3
9,7	8,4	9,7	8,9
5,1	6,1	5,1	7,8
6,2	8,2	6,2	5,2
5,4	9,0	5,4	9,4
6,5	8,5	6,5	7,6
8,7	5,4	8,7	8,2
6,2	5,2	6,2	5,1
8,3	5,1	8,3	8,8
9,5	5,0	9,5	6,3
5,9	5,0	5,9	8,9
5,0	5,1	5	7,5
5,3	5,9	5,3	7,1
6,4	5,6	6,4	5,3
5,0	5,2	5	8,8
6,0	5,1	6	9,2
6,0	6,6	6	8
5,5	6,7	5,5	9
9,7	5,9	9,7	6
9,4	5,0	9,4	5,1
5,3	7,3	5,3	8
5,9	6,0	5,9	6,3
9,2	5,1	9,2	6
8,1	5,0	8,1	8,2
5,5	5,0	5,5	7,6
9,7	5,1	9,7	5
5,3	7,2	5,3	9,4
9,4	9,5	9,7	5,8
5,1	6,8	5,1	5,6
9,5	5,1	9,5	5,1
5,3	7,6	5,3	6
5,3	9,2	5,3	5
8,8	5,0	7,5	8,6
7,4	6,4	8,8	8,5
5,5	6,4	7,4	7
7,8	5,1	5,5	6,6
5,5	5,3	7,8	9
8,5	7,0	5,5	6,3
9,5		8,5	7,8
9,2		9,5	7,6
6,6		9,2	8,1
8,5		6,6	7,1
8,3		8,5	7,9
8,7		8,3	5,1
5,5		5,7	7,4
5,0		5,5	9,5
10,0		5	7,3
4,4		10	6,5
9,2		7,4	5,3
9,2		9,2	7,4
9,5		9,2	5
6,9		9,5	7,3
8,0		6,9	9,2
5,7		8	5,9
5,0		5,7	7,5
7,8		5	5,2
6,2		7,8	8
5,7		6,2	7
8,3		5	7,5
5,0		6,9	
8,7		5,7	
5,8		8,3	
6,5		5	
5,3		8,7	
8,8		5,8	
7,8		6,5	
		5,3	
		8,8	
		7,8	