

Uma Experiência na Aplicação de Práticas de Apoio no Ensino-Aprendizado de Algoritmos

Gabriel Costa Silva¹, Reginaldo Ré¹, André Kawamoto¹, André Schwerz¹

¹Coordenação do Curso Superior de Tecnologia em Sistemas para Internet – Universidade Tecnológica Federal do Paraná (UTFPR)

Caixa Postal 271 – 87301-006 – Campo Mourão – PR – Brasil

{gabrielcosta, reginaldo, kawamoto, andreluis}@utfpr.edu.br

Abstract. *The algorithms teaching is a challenge for the teachers from computing courses. Unlike other technical subjects, algorithms not only requires that students to learn technology, but it aims to develop a set of skills, as: (i) logical reasoning (ii) the ability to solve problems; and (iii) the ability to abstracting solutions using a formal representation. This paper describes an experience using few practices to support the learning-teaching process. These practices aim: (i) minimizing the algorithmic abstraction; (ii) identify learning weaknesses in order to solve them on time; and (iii) motivate the student through ludic problems.*

Resumo. *O ensino de algoritmos é um desafio para professores de cursos de computação. Diferentemente de algumas disciplinas técnicas, algoritmos exige do aluno não apenas o aprendizado de uma tecnologia, mas sim um conjunto de habilidades como: (i) raciocínio lógico; (ii) capacidade de resolver problemas; e (iii) capacidade de abstrair soluções usando uma representação formal. Neste artigo é relatada uma experiência na aplicação de práticas de apoio no processo de ensino-aprendizado de algoritmos usando diferentes ambientes de apoio à aprendizagem. Tais práticas visam: (i) reduzir a abstração de um algoritmo; (ii) identificar deficiências no aprendizado e atacá-las a tempo; e (iii) motivar o aluno por meio de problemas lúdicos.*

1. Introdução

É comum alunos de cursos da área de computação encontrarem dificuldades na disciplina de algoritmos. No entanto, a disciplina é também um desafio para os professores [Nakamura 2001]. Nessa disciplina o professor deve ensinar o aluno a resolver problemas usando uma representação formal. Isso exige do aluno uma série de habilidades como: (i) interpretação de texto; (ii) raciocínio lógico; (iii) bom uso da matemática; (iv) uso de estruturas algorítmicas; e (v) habilidade de programação. Algumas ferramentas e metodologias têm sido propostas para auxiliar o professor a oportunizar o desenvolvimento dessas habilidades para os alunos [Jesus e Raabe 2010; Vahldick e Mattos 2008]. No entanto, deve-se considerar que assumir a disciplina e aplicar tais propostas de modo que gerem resultados concretos não é uma tarefa trivial.

Neste artigo são relatadas três práticas aplicadas como apoio no processo de ensino-aprendizado de algoritmos. Este artigo está organizado da seguinte forma: na Seção 2 a disciplina de algoritmos é apresentada. Na Seção 3 são relatadas as práticas aplicadas com suas respectivas ferramentas de apoio. Também são apresentados os primeiros resultados apurados após a aplicação de cada prática. Por fim, na Seção 4 são apresentadas as conclusões considerando o conjunto das práticas e o avanço dos alunos.

2. Algoritmos, a Disciplina e seus Desafios

Cormen et al. (2002) definem algoritmos como um procedimento computacional que transforma um valor de entrada em uma saída. Para apoiar tal procedimento, são usadas: (i) estruturas de controle; e (ii) estruturas de dados. As Diretrizes Curriculares de Cursos da Área de Computação e Informática [MEC 1999] posicionam algoritmos na área de formação básica, entendendo que a habilidade em algoritmos é necessária para o desenvolvimento tecnológico na área da computação. A *Association for Computing Machinery* (ACM) e o *Institute of Electrical and Electronics Engineers* (IEEE) também ressaltam essa importância em seu currículo para cursos de computação [CS2008 2008]. Tal importância é justificável dada a relação da disciplina com as demais disciplinas de um curso da área de computação. Assim, um fraco desempenho em algoritmos compromete todo o curso. A literatura cita diversos fatores para o fracasso dos alunos nessa disciplina. Três deles são comuns na maioria dos trabalhos: (i) forte carga de conceitos abstratos [Júnior e Freitas 2010]; (ii) falhas no aprendizado [Ynoguti 2005]; e (iii) falta de motivação [Vahldick e Mattos 2008; Jesus e Raabe 2010].

3. Experiências

As práticas de apoio foram aplicadas à turma ingressante no curso superior de tecnologia em Sistemas para Internet da Universidade Tecnológica Federal do Paraná – câmpus Campo Mourão (UTFPR-CM) no primeiro semestre de 2011. O professor que aplicou as práticas é o mesmo que ministrou a disciplina no segundo semestre de 2010, assim os resultados obtidos podem ser diretamente comparados com a turma anterior – que não teve as práticas. A linguagem de programação Java é usada na disciplina.

Cada prática foi apoiada por ferramentas de apoio ao ensino: (i) JELiot, usado para tentar reduzir a abstração computacional; (ii) Moodle, usado para analisar o aproveitamento do aluno em cada aula; e (iii) Furbot [Vahldick e Mattos 2008], usado para motivar os alunos por meio da prática de exercícios lúdicos. Apesar de existirem outras ferramentas que possam ser usadas com o mesmo propósito, essas ferramentas foram escolhidas por serem: (i) estáveis; (ii) relativamente simples de configurar e aplicar; (iii) compatíveis com a linguagem de programação usada no curso; e (iv) estarem em português. As seções seguintes relatam a aplicação das práticas. Os resultados apresentados nas seções foram apurados por meio de questionários aplicados aos alunos (JELiot e Furbot) ou por meio de notas obtidas (Moodle).

3.1. Visualização da Abstração Computacional

Durante a execução do algoritmo o indivíduo insere dados e obtém resultados. Assim, comparando as entradas e saídas de dados um aluno pode concluir se o seu algoritmo fez o que era esperado. No entanto, se houver algum problema na lógica do algoritmo o aluno pode obter resultados corretos para alguns valores de entrada, mas não para todos. Como o processo de execução do algoritmo é visualizado de forma atômica pelo aluno, ele não consegue acompanhar os passos da execução e, conseqüentemente, tem dificuldades para entender e localizar erros em seu algoritmo. O JELiot é um editor de código Java que permite acompanhar a execução do algoritmo a cada instrução, bem como os processos computacionais que são simulados em uma espécie de cenário.

O JELiot foi apresentado aos alunos como seu primeiro editor de código. Nas primeiras aulas foi explicado seu funcionamento e, a partir de então, todas as atividades práticas passaram a adotar o JELiot. O fato do programa estar em inglês pareceu um problema no início, mas os alunos se adaptaram bem. Em alguns casos, o nível de detalhes na simulação também gerava certa confusão. Para analisar o ponto de vista dos alunos em relação a esta prática, um questionário foi aplicado para um grupo de 22 alunos. Como resultado, 95% consideraram o JELiot útil para visualizar o que acontece no algoritmo. Quando questionados sobre preferência em relação aos editores, 59% afirmaram preferir o JELiot. Os outros 9 alunos preferiram outro editor. Acredita-se que isso se deva às facilidades oferecidas por outros

editores, como a funcionalidade de auto-completar o código e a identificação de erros em tempo de codificação. Dos alunos pesquisados, 81% consideram a simulação do JELiot clara o suficiente para entender seu algoritmo. Por fim, quando questionados sobre o benefício do JELiot, 68% afirmam que sempre ou quase sempre o JELiot os ajuda a entender melhor o funcionamento do algoritmo. Um ponto a ser ressaltado é que, nessa última pergunta, nenhum aluno marcou a opção 'Raramente' ou 'Nunca'.

3.2. Verificação de Aprendizado

Os resultados dos trabalhos e das provas obtidos durante o semestre mostram o aproveitamento da turma em relação aos conteúdos ministrados. No entanto esses resultados podem ser apurados tarde demais. A Verificação de Aprendizado (VA) visa identificar dificuldades do aluno ainda em tempo de corrigi-las. A VA é um questionário que contém de 3 a 5 questões. A VA é aplicada pela Internet usando o Moodle. O Moodle é um sistema de gerenciamento de cursos que permite conduzir cursos total ou parcialmente virtuais. Por meio do Moodle é possível disponibilizar material, aplicar provas, estabelecer fóruns e receber atividades de alunos, entre outros. Para cada VA o aluno tem 3 chances de responder o questionário. Quando o aluno acerta, ele recebe uma parabenização, junto com a justificativa do acerto. No caso de erro, o aluno é informado e recebe uma dica para uma nova tentativa. Cada VA é analisada pelo professor antes da próxima aula, assim é possível identificar as deficiências da turma antes que um novo conteúdo seja abordado.

O processo de elaboração da VA é trabalhoso e demorado. As questões não devem deixar margem para dupla interpretação, uma vez que a VA é respondida fora da sala de aula. Além disso, para permitir a autocorreção do sistema, as questões devem ser do tipo objetiva. O cadastramento das questões é feito de forma simples e intuitiva, no entanto, toma um tempo considerável. A elaboração das questões, possíveis respostas e dicas também são um desafio. Por fim, a análise dos dados no relatório de resultados no Moodle não é simples. Uma vez feito isso, deve ser pensado como os problemas encontrados serão atacados. Normalmente isso é feito por meio de uma rápida revisão dos conceitos, discussão da questão da VA em sala e aplicação de um novo exercício relacionado. Outro desafio é conscientizar o aluno a usar a VA da forma correta: (i) estudar o conteúdo; (ii) responder o questionário; (iii) analisar erros e acertos; e (iv) repetir o processo até esgotar as tentativas ou a nota máxima ser atingida.

Uma análise estatística considerando 5 atividades de VA aplicadas antes da primeira prova oficial mostrou que tanto a média quanto a moda das tentativas é de 2 tentativas a cada VA. A nota média obtida na primeira tentativa, em uma escala de 0 a 10, é de 5.5, enquanto na última tentativa é de 8.8, um crescimento de 60%. Apesar da moda da nota na última tentativa ser 10, a mediana é de 9.8. Uma questão intrigante é que, apesar das boas notas na VA as notas das provas foram bem menores. As provas mantiveram a mesma complexidade e estilo das VA.

3.3. Motivação por meio de Problema Lúdico

Soluções lúdicas são propostas como meio para motivar o aluno, considerando que o aspecto lúdico é um importante fator motivador [Jesus e Raabe 2010]. O Furbot é um *framework* que cria um ambiente lúdico no qual reside um robô e, opcionalmente, outros elementos. O comportamento do robô é definido pelo aluno usando instruções do Furbot e estruturas algorítmicas implementadas em Java.

O programa foi apresentado aos alunos no segundo mês de aula, após eles terem aprendido a usar estruturas de seleção. Na primeira aula foram feitas demonstrações e, em seguida, os alunos receberam tarefas simples, como fazer o robô andar pelo ambiente e desviar de elementos. A partir de então, para cada nova estrutura algorítmica ensinada eram alocadas aulas para exercícios com o Furbot. A criação das tarefas e configuração do ambiente não são atividades complexas, mas exigem criatividade para aplicar problemas clássicos em um novo cenário. As instruções são intuitivas, além disso existe uma

pouca quantidade de instruções que são úteis para a maioria dos problemas. O código é estável e funcionou sem problemas nos sistemas testados.

Uma pesquisa realizada com um grupo de 21 alunos analisou a apreciação, complexidade, motivação e benefício do uso do Furbot no ponto de vista dos alunos. Quando questionados sobre o nível de apreciação que eles têm no uso do Furbot, 76% classificaram como médio, dizendo que “acham legal usar”. Um aluno indicou que sente repúdio. Em relação à complexidade, 71% classificaram o Furbot como tendo um nível médio. No questionamento sobre motivação, 28% indicaram que se sentem mais motivados a aprender algoritmos usando o Furbot. Por fim, os alunos foram questionados se os exercícios usando o Furbot ajudam a entender melhor os conceitos de algoritmos. Dos alunos questionados 52% indicaram que sim.

4. Conclusões

Abstração, assuntos correlatos e falta de motivação dos alunos são apontados como os principais responsáveis pelos altos níveis de reprova e desistência na disciplina de algoritmos. Assim, nesse artigo é relatada uma experiência na qual são aplicadas 3 práticas de apoio ao ensino-aprendizado visando atacar tais problemas. As práticas usam diferentes ferramentas de apoio. Apesar de, quando analisadas individualmente considerando o ponto de vista dos alunos, apenas uma prática ter atingido seu objetivo, quando analisadas em conjunto, as práticas revelam resultados interessantes. Considerando o semestre anterior, o número de desistências caiu de 58% para 28%, quando comparados os números de alunos ingressantes e concluintes. Já o número de alunos aprovados subiu 80%. Considerando que as únicas mudanças ocorridas na disciplina foram a aplicação dessas práticas, pode-se sugerir que tais práticas geraram resultados positivos para o curso.

Referências

- Cormen, T.H., Leiserson, C.E., Rivest, R.L., Stein, C. (2002) Algoritmos: teoria e prática, Campus/Elsevier, 12 ed.
- CS2008 – Computer Science Curriculum Update (2008) Disponível em: <<http://www.acm.org/education/curricula/ComputerScience2008.pdf>>, Acesso em 02 ago. 2011.
- Jesus, E. A. de, Raabe, A. L. A. (2010) Avaliação Empírica da Utilização de um Jogo para Auxiliar a Aprendizagem de Programação, In: *Anais do XXI Simpósio Brasileiro de Informática na Educação*, João Pessoa.
- Júnior, D. P., Freitas, R. L. (2010) Estratégias para melhorar os processos de abstração na disciplina de Algoritmos, In: *Anais do XXI Simpósio Brasileiro de Informática na Educação*, João Pessoa.
- MEC – Ministério da Educação. Diretrizes Curriculares para Cursos da Área de Computação e Informática (1999) Disponível em: <<http://www.inf.ufrgs.br/mec/ceeinf.diretrizes.html>>, Acesso em 9 jun. 2011.
- Nakamura, A. S. (2001) Um estudo sobre o ensino da lógica de programação em cursos superiores de informática. 163p. Tese (Doutorado) - Universidade Estadual Paulista Júlio De Mesquita Filho, Marília.
- Vahldick, A., Mattos, M. M. (2008) Relato de uma Experiência no Ensino de Algoritmos e Programação Utilizando um Framework Lúdico, In: *Anais do II Workshop de Ambientes de apoio à Aprendizagem de Algoritmos e Programação*, Fortaleza.
- Ynoguti, C. A. (2005) Uma metodologia para o ensino de algoritmos, In: *Global Congress on Engineering and Technology Education*, pp. 684-687.