
Andreia de Jesus²
Gláucia Silva Brito³

**CONCEPÇÃO DE ENSINO-APRENDIZAGEM
DE ALGORITMOS E PROGRAMAÇÃO DE
COMPUTADORES: A PRÁTICA DOCENTE¹**

RESUMO: O objetivo deste artigo é apresentar uma visão geral sobre a concepção de ensino-aprendizagem de algoritmos e programação de computadores, pontuar as principais dificuldades enfrentadas pelos alunos e apresentar o perfil dos professores, tendo como base literaturas relacionadas com o assunto. Após a conclusão da análise destes itens, propõe-se uma metodologia de plano de aula com base na taxionomia de Bloom, visando auxiliar o professor no processo ensino-aprendizagem de algoritmos e programação de computadores.

PALAVRAS-CHAVE: metodologia, plano de aula, taxionomia de Bloom.

TEACHING AND LEARNING CONCEPTION OF ALGORITHMS AND
COMPUTER PROGRAMMING: THE TEACHING PRACTICE

SUMMARY: The objective of this paper is to show an overview of conceptions on teaching and learning algorithms and computer programming to point out the main difficulties students deal with and present the profile of teachers based on literature related to the topic. After this, it is proposed a methodology to help teachers to prepare their syllabus based on Bloom's taxonomy.

KEYWORDS: methodology, syllabus, Bloom's taxonomy.

Data de recebimento: 22/06/2009. Data de aceite para publicação: 29/09/2009.

¹ Trabalho apresentado no I Encontro Nacional de Informática e Educação, Cascavel 2009.

² Cientista da Computação, Coordenação do curso de Sistemas de Informação, Faculdades Integradas do Brasil, Curitiba, PR, CEP 82820-540, sistemas@unibrasil.com.br.

³ Professora Adjunta da Universidade Federal do Paraná, Curitiba, PR

INTRODUÇÃO

Disciplinas como algoritmo e programação de computadores exigem dos alunos habilidades e competências como: raciocínio lógico, resolução de problemas e a capacidade de abstração da solução em uma representação formal e/ou em uma linguagem computacional.

Autores como RAABE e SILVA (2005) citam a abstração como um dos principais fatores da desmotivação e do elevado grau de desistência dos alunos. Além disso, essas habilidades e competências são pré-requisitos para a maioria das disciplinas trabalhadas em cursos de Computação e Informática. Portanto, é primordial que sejam desenvolvidos aplicativos e pesquisadas metodologias que venham, efetivamente, contribuir para a melhoria da qualidade do processo de ensino-aprendizagem destes alunos.

Em geral, os professores utilizam exemplos hipotéticos para demonstrar o comportamento da máquina (computador) no momento da execução das soluções abstraídas em linguagem computacional. Mas isto não é suficiente, pois os alunos não conseguem visualizar efetivamente como as mudanças de estado ocorrem no equipamento. Por isso, pesquisas estão sendo realizadas e ferramentas computacionais estão sendo desenvolvidas com este objetivo por RAABE e SILVA (2005), NOBRE e MENEZES (2002), DELGADO et al. (2004), entre outros. Vale ressaltar que tanto as metodologias, como os aplicativos educacionais devem ser desenvolvidos de tal forma que durante a sua utilização os alunos se sintam motivados e consigam ter nessas ferramentas um auxílio a mais no desenvolvimento das competências necessárias para a construção de algoritmos e programação de computadores.

Portanto, é de suma importância que os professores que lecionam algoritmos e programação de computadores tenham uma definição bem clara da concepção do processo de ensino-aprendizado desses conteúdos, para que os mesmos possam desenvolver metodologias que realmente contribuam para o desenvolvimento das competências e habilidades exigidas, além de fazerem uso adequado de aplicativos destinados a este fim.

DISCIPLINAS DE ALGORITMOS E PROGRAMAÇÃO DE COMPUTADORES

Para se compreender a natureza das disciplinas de algoritmos e programação de computadores é preciso, primeiramente, conceituar três termos bastante utilizados nestas disciplinas: lógica, algoritmos e linguagem de programação.

Para FORBELLONE (2000, p. 1) a lógica é a “arte do bem pensar”.

O raciocínio é a forma mais complexa do pensamento e a lógica estuda a correção do raciocínio, colocando coerência e ordem no pensamento (Idem). Ainda ele também define o que é lógica de programação, descrevendo que além de visar a ordem da razão, esta também se preocupa com a simbolização formal do raciocínio para a programação dos computadores. Já o algoritmo é uma sequência lógica de passos que visa atingir um objetivo bem definido (Idem). Por fim, as linguagens de programação constituem ferramentas para a implementação de software.

Com as definições apresentadas é possível fazer a seguinte relação: para se construir um algoritmo é preciso primeiramente desenvolver o raciocínio lógico, pois esta habilidade é fundamental nesta tarefa. Além disso, é preciso compreender o funcionamento das linguagens de programação, quanto à sintaxe e à semântica, para que seja possível traduzir uma solução algorítmica para um programa de computador. Logo, a construção de algoritmos é uma das várias etapas da atividade de programação. DELGADO et al. (2005) identificou, em seu trabalho de pesquisa, as subcompetências para a atividade de construção de algoritmos: (1) Interpretação e compreensão de texto; (2) Resolução de problemas; (3) Formalização da solução proposta; (4) Construção de algoritmos.

Outra característica importante dos algoritmos é que não existe uma única solução para um determinado problema. Como é necessário aplicar lógica na construção de um algoritmo e/ou programa, a solução é bastante subjetiva, pois o raciocínio lógico é particular de cada pessoa. Logo, para um determinado problema é possível apresentar várias soluções com caminhos diferentes, mas todos serão capazes de alcançarem o mesmo resultado. Devido a isto, é necessário saber formalizar as soluções propostas de forma a abstraí-las em diversos níveis e permitir assim uma compreensão universal da solução. Com o Quadro 1 é possível visualizar as diversas abstrações para um determinado problema e, com isso, verificar que quanto mais alto o nível de abstração, mais a solução se aproxima da linguagem computacional. Além disso, o quadro possibilita reforçar que a formalização, independente do nível, se preocupa com o processo de solução e não mais com a solução em si.

Considerando que a tarefa de construção de algoritmos é bastante complexa, é necessário que diversas atividades sejam desenvolvidas para que o aluno seja capaz de compreender soluções algorítmicas prontas, de propor suas próprias soluções e de realizar relações entre diferentes soluções para um mesmo problema.

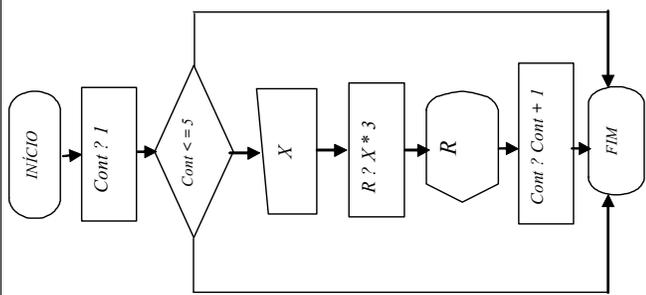
PERFIL DE PROFESSORES E ALUNOS

Estudos como os de RAABE e SILVA (2005) coloca que professores não estão preparados para ensinar resolução de problemas aos alunos e também têm dificuldades de compreender a lógica desses alunos. Em geral, a teoria é apresentada através de modelos prontos, ou seja, através de técnicas, estratégias e soluções do próprio professor. Isso leva o aluno a se deparar com um problema a ser solucionado e uma solução a ser formalizada, a reproduzir o que foi apresentado em sala. Neste momento esse aluno tem a percepção de que ele entendeu a solução apresentada pelo professor, mas que não compreendeu como esse processo se desenvolveu. Isto dificulta a habilidade desse aluno de desenvolver suas próprias soluções, pois o que lhe foi passado foi um modelo de algoritmo pronto e não o processo de desenvolvimento deste modelo.

Tendo em vista o que foi descrito, percebe-se que nas disciplinas de Algoritmos e Programação de Computadores há uma forte ênfase no ensino da construção de algoritmos e programas computacionais, mas que as atividades de interpretação e compreensão de texto, resolução de problemas e formalização da solução proposta, que são habilidades essenciais para a compreensão algorítmica, estão sendo deixadas de lado ou em segundo plano neste processo de ensino.

Em geral, os alunos não sabem por onde começar quando um problema lhe é apresentado e uma solução para tal lhe é solicitada. Isto se deve, principalmente, pelo fato de que os alunos apresentam dificuldades na interpretação do enunciado do problema. Eles não conseguem identificar no texto quais são as variáveis de entrada, o que precisa ser processado e quais são as variáveis de saída. Ou seja, ter conhecimento da sintática e semântica das linguagens de programação não é suficiente se os alunos não conseguem compreender e propor soluções para os problemas, etapa esta que antecede a construção de algoritmos e programas computacionais. Mas, também, como coloca BRANCO NETO & CECHIMEL (2006), a simples ignorância dos aspectos sintáticos básicos também dificulta o aprendizado e a motivação dos alunos para a programação de sistemas computacionais.

Além disso, LISTER & LEANY (2003) apresentam que os alunos mais preparados não são desafiados pelo professor, enquanto os mais fracos não conseguem ter suas deficiências identificadas e compreendidas por esse profissional que encontra dificuldades para esse reconhecimento. Esta atitude contribui para o grande número de evasão dos alunos dessas disciplinas e, conseqüentemente, dos cursos da área de computação. Tendo isto conhecido, os autores colocam a necessidade de fazer uso de uma metodologia de ensino-aprendizagem

<p>Enunciado do problema: Desenvolver um programa que lê um valor para a variável X, multiplica este valor por 3, armazena o resultado em um variável R e imprime o conteúdo desta variável. O programa deve repetir 5 vezes esta sequência de comandos.</p>	<p>Abstração 1 Linguagem Natural</p>	<p>Abstração 2 Representação Gráfica</p> 	<p>Abstração 3 Linguagem Algorítmica</p>	<p>Abstração 4 Linguagem de Programação C</p>
<ol style="list-style-type: none"> 1. Criar uma variável para ser o contador iniciando em 1; 2. Enquanto o valor do contador não for 5, executar os passos 3, 4 e 5; 3. Ler um valor para a variável X; 4. Efetuar a multiplicação do valor de X por 3, armazenando o resultado em R; 5. Apresentar o conteúdo de R; 6. Acrescentar mais um valor na variável contador; 7. Quando o contador for maior do que 5, encerrar o processamento repetição (looping). 			<p>Algoritmo Looping Var: X, R, I: Inteiro; Início I Enquanto (I <= 5) Leia X; R Escreva R; I Fim-Enquanto; Fim</p>	<pre>include<studio.h> #include<conio.h> void main() { int X, R, I=1; clrscr(); while(I <= 5) { printf("Digite um valor inteiro:"); scanf("%d", &X); R = X * 3; printf("%d", R); I++; } }</pre>

Quadro 1- Níveis de abstração de um algoritmo.

que atinja todos os níveis de conhecimento e habilidades dos alunos. Para isso, eles aplicam a chamada taxionomia de Bloom para propor uma metodologia de ensino-aprendizado e avaliação desses conteúdos. A taxionomia de Bloom possui seis níveis, sendo que os níveis subsequentes dependem das habilidades desenvolvidas nos níveis anteriores. Os níveis em ordem crescente são os seguintes: conhecimento, compreensão, aplicação, análise, síntese e avaliação. Com esta proposta os alunos são ensinados a analisar uma descrição vaga de um problema, refiná-lo, decompô-lo em partes e determinar métodos apropriados para cada parte. A solução do problema, neste caso, é a tarefa em nível de síntese.

PROPOSTA METODOLÓGICA DE PLANO DE AULA PARA ALGORITMOS E PROGRAMAÇÃO DE COMPUTADORES

Tendo como base a concepção de ensino-aprendizagem de algoritmos e programação de computadores e as dificuldades de aprendizado dos alunos, segue uma proposta de plano de aula que visa auxiliar o professor no processo de ensino de algoritmos e programação de computadores.

Esta proposta de plano de aula tem como 1ª etapa a resolução de problemas. O objetivo é desenvolver as habilidades referentes às fases 1 e 2 da metodologia proposta por DELGADO et al. (2004). É válido ressaltar que a fase 1 não trata de algoritmos e programação e sim de técnicas para a resolução de problemas. E a fase 2 busca a formalização, em linguagem natural, do método utilizado para a solução, além da prática de resolução de problemas mais complexos.

A 2ª etapa do plano de aula proposto tem como objetivo abordar a terceira fase da metodologia de DELGADO et al. (ibidem), que é a construção de algoritmos. Nesta fase é preciso trabalhar os conceitos das diversas estruturas computacionais necessárias para a prática de programação de computadores, bem como a sintaxe utilizada para transcrever a solução do problema em uma linguagem algorítmica ou computacional.

Nesta fase de construção de algoritmos deverão ser trabalhados os conceitos de cada estrutura de forma isolada. Para cada conceito abordado o professor deverá definir atividades que possibilitam o aluno a desenvolver os níveis de conhecimento, compreensão e aplicação da taxonomia de Bloom. Além disso, com base nestas categorias, cabe ao professor selecionar aplicativos que possam ser utilizados para apoiar o processo ensino-aprendizagem em cada um desses níveis. Para esta seleção propõe-se aplicar a metodologia apresentada por JESUS (2008) que integra vários tipos de classificação de softwares: tipo geral de

software, ciclo de desenvolvimento, tipo de ambiente de aprendizado, o nível de aprendizado, tipo de ferramenta para o ensino de algoritmos e pela classificação dos objetivos de aprendizado.

Para a elaboração das aulas nesta segunda etapa propõe-se um roteiro que determina a abordagem do objetivo de aprendizado de cada conceito e suas respectivas atividades seja no nível de conhecimento, de compreensão e de aplicação, bem como a aplicação desses conceitos nos níveis determinados. Em nível de conhecimento (com o uso de verbos como citar, nomear, listar, reproduzir, repetir e apresentar, que refletem este nível) deve ser abordada a definição do conceito apresentado, bem como a expressão sintática que expressa o conceito no algoritmo ou programa. Nesse nível exemplos devem ser apresentados aos alunos.

Já em nível de compreensão (usando verbos como explicar, relacionar, traduzir, transformar, descrever e associar) os alunos devem ser estimulados a explicar o funcionamento lógico do conceito estudado, bem como identificar este conceito em solução de problemas já modelados na linguagem natural. No nível de aplicação (com verbos tais como aplicar, prever, demonstrar, preparar, resolver e modelar) os alunos devem ser estimulados a resolver problemas que necessitam do conceito estudado para a sua solução.

Aos professores fica a proposta de demonstrar aos alunos como a aplicação do conceito pode melhorar um algoritmo ou programa que já foi desenvolvido por eles e como o conceito, se aplicado adequadamente a uma determinada solução, pode contribuir para que o algoritmo, quando implementado em uma linguagem de programação, exija menos recursos da máquina.

Na sequência é importante definir aplicativos que possam ser utilizados como apoio no processo de desenvolvimento de cada nível. Neste ponto é importante que o professor realize uma análise do(s) aplicativo(s) proposto(s) para identificar, conforme metodologia proposta por JESUS (2008), em qual ou quais dos níveis da taxonomia de Bloom a ferramenta(s) poderá(ão) ser utilizada(s) como apoio no processo.

Para exemplificar a aplicação do roteiro proposto, será tomado um conceito operacional de programação modular, que é o de Passagem de Parâmetro.

Em nível de Conhecimento têm-se as seguintes indicações: (1) Objetivos de Aprendizado: definir o conceito de passagem de parâmetro por valor e por referência, suas respectivas sintaxes e conceitos intrínsecos a sua utilização, como variável global e local. (2) Atividades Propostas: identificar, segundo a sintaxe apresentada nos algoritmos e/ou programas, se é uma passagem de parâmetro por valor ou por referência e corrigir a sintaxe dos algoritmos e/ou programas no que diz respeito à passagem de parâmetro. Este exercício deve reforçar os conceitos de variável global e local. (3) Definir Aplicativo: através da

análise e classificação do ambiente A4, realizada por JESUS (2008), foi possível identificar que este ambiente auxilia o desenvolvimento em nível de conhecimento através do tutorial e da lista de links interessantes. Além disso, o editor de algoritmos permite que os alunos corrijam a sintaxe de algoritmos pré-definidos e editem os seus próprios algoritmos, os quais podem ser submetidos ao professor para a correção da sintaxe. E por tratar de um ambiente distribuído socialmente possibilita, através do chat e do fórum, a socialização e discussão dos conteúdos em momentos extra classe.

Em Nível de Compreensão as indicações podem ser ordenadas por: (1) Objetivos de Aprendizado: discutir com os alunos quando e por que a passagem de parâmetro deve ser utilizada. Quais as vantagens que este conceito traz para a atividade de programação de computadores. (2) Atividades Propostas: descrever a lógica de funcionamento da passagem de parâmetro por valor e por referência. Esta descrição deve ser socializada com os demais colegas de turma para analisar as diferentes percepções, o professor deve propor aos alunos que traduzam as soluções de problemas já formalizadas na linguagem natural para a linguagem algorítmica, utilizando o conceito de passagem de parâmetro e também apresentar aos alunos programas que utilizam conceito de passagem de parâmetro e que apresentam erros de lógica. Os alunos devem debugar (executar passo a passo) estes programas para identificar o problema e propor solução para eles. (3) Definir Aplicativos: neste nível de aprendizado duas ferramentas, segundo JESUS (2008), podem apoiar o processos sendo a primeira delas o Ambiente A4, que além de proporcionar ao aluno a socialização e a discussão sobre a sua compreensão do conceito, o possibilita a continuar utilizando o tutorial e os links interessantes para estar pesquisando novos exemplos e reforçando a sintaxe e o simulador de algoritmos WebPortugol que os possibilita debugar os programas propostos. Esta prática proporciona uma melhor compreensão, por parte do aluno, sobre o comportamento da máquina quando um determinado conceito é aplicado e executado. Além de facilitar a compreensão de erros de lógica relacionados a um determinado conceito.

Em nível de Aplicação as sugestões seguem em: (1) Objetivos de Aprendizado: discutir com os alunos quando e por que aplicar este conceito para a resolução de problemas computacionais. (2) Atividades Propostas: o professor deve apresentar aos alunos diversos enunciados de problemas e solicitar que eles identifiquem quando e porque a passagem de parâmetro deve ser aplicada. Ele também deve solicitar que os alunos apresentem suas próprias soluções algorítmicas com passagem de parâmetro, bem como demonstrar como a passagem de parâmetro pode melhorar a organização de algoritmos e programas já desenvolvidos por eles para com isso contribuir para o

reaproveitamento de código. É importante que nesta atividade sejam utilizados algoritmos que os alunos já desenvolveram, pois desta forma eles irão direcionar a atenção para o conceito que está sendo abordado e não para uma solução de problemas que ainda não estudaram. Ainda cabe ao professor modelar, junto com os alunos, situações em que o uso de parâmetros diminuirá o número de comandos (linhas) de um algoritmo e/ou programa, ou diminuirá o número de variáveis e, com isso, exigirá menos recurso da máquina quando implementado em uma linguagem de programação. (3) Definir Aplicativo: de acordo com JESUS (2008) o ambiente A4 poderá ser utilizado nesta fase como uma ferramenta de apoio para socializar e discutir as demonstrações e modelagens trabalhadas em sala de aula e para os alunos pesquisarem nos links interessantes outras aplicações de passagem de parâmetro. Com o simulador WebPortugol os alunos poderão implementar as suas soluções e debugá-las. Isto fará com que os alunos pensem sobre suas próprias ideias, reavaliando a sua percepção sobre o que é passagem de parâmetro e por que utilizá-la.

Após trabalhar o nível de conhecimento, compreensão e aplicação de cada conceito os professores devem ter como objetivo desenvolver nos alunos o nível de análise, síntese e avaliação. No caso da análise e síntese é necessário propor atividades que instiguem os alunos a analisar como cada conceito se comporta individualmente e como diferentes conceitos se relacionam na solução de um problema único. Nesta fase, os problemas propostos devem exigir para as suas soluções a integração de diferentes conceitos. Desta forma o aluno passa a ter uma visão macro e não micro das estruturas de programação. No que diz respeito à avaliação deve-se propor aos alunos que façam julgamentos sobre as soluções propostas, para verificar a exatidão, a efetividade, a economia de recursos computacionais e/ou a satisfação da atividade proposta.

CONCLUSÃO

É de suma importância que professores que lecionam algoritmos e programação de computadores tenham uma definição bem clara da concepção de ensino-aprendizagem desses conteúdos, para que os mesmos possam desenvolver metodologias que realmente contribuam para o desenvolvimento das competências e habilidades exigidas, além de fazerem uso adequado de aplicativos destinados a este fim.

Logo o objetivo deste artigo foi apresentar uma visão geral sobre a concepção de ensino-aprendizagem de algoritmos e programação de computadores, pontuar as principais dificuldades enfrentadas pelos alunos e apresentar o perfil dos professores, tendo como base literaturas relacionadas com o assunto. Como conclusão da análise destes itens

foi proposta uma metodologia de plano de aula que visa auxiliar o professor no processo de ensino-aprendizagem de algoritmos e programação de computadores.

Um próximo passo é validar a proposta na prática, pois desta forma será possível avaliar a sua eficácia e propor melhorias no processo. Além disso, há diferenças de perfis entre alunos e entre turmas e isto, geralmente, implica em adaptações das propostas de ensino-aprendizagem de acordo com as características de seu público alvo.

REFERÊNCIAS

BRANCO NETO, W. C.; CECHINEL, C. Uma análise dos problemas enfrentados no ensino-aprendizagem de fundamentos de programação à luz da taxionomia de Bloom. In: XIV WORKSHOP SOBRE EDUCAÇÃO EM COMPUTAÇÃO, 2006.

DELGADO, C., XEXEO, J. A. M., SOUZA, I. F., CAMPOS, M., RAPKIEWICZ, C. E. Uma abordagem pedagógica para a iniciação ao estudo de algoritmos. In: XII WORKSHOP DE EDUCAÇÃO EM COMPUTAÇÃO (WEI'2004), 2004, Salvador, BA.

DELGADO, C., XEXEO, J. A. M., SOUZA, M., RAPKIEWICZ, C. E., Junior, J.C.P. Identificando competências associadas ao aprendizado de leitura e construção de algoritmos. In: XXV SBC, 2005, Unisinós, São Leopoldo, RS.

FORBELLONE, A. L. V.; EBERSPÄCHER, H. F.. **Lógica de programação**. 2 ed. São Paulo: Makron Books, 2000. ISBN: 85.346.1124-6.

JESUS, A. **O processo de ensino-aprendizado de algoritmos e programação**: Uso de aplicativos. Tese (Especialista em Informática na Educação) - Instituto Brasileiro de Pós-Graduação e Extensão – IBPEX, 2008.

LISTER, R.; LEANY, J. First year programming: Let all the flowers bloom. In: FIFTH AUSTRALASIAN COMPUTING EDUCATION CONFERENCE (ACE2003), 2003, Adelaide, Australia.

NOBRE, I. A. M. N.; MENEZES, C. S. Suporte à cooperação em um ambiente de aprendizagem para programação (SambA). In: XIII SIMPÓSIO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO – SBIE, 2002. São Leopoldo, RS, 2002.

RAABE, A. L. A.; SILVA, J. M. C. da. Um ambiente para atendimento às dificuldades de aprendizagem de algoritmos. In: XIII WORKSHOP SOBRE EDUCAÇÃO EM COMPUTAÇÃO, 2005, São Leopoldo, RS, 2005.