

UNIVERSIDADE CANDIDO MENDES
PÓS-GRADUAÇÃO “LATO SENSU”
AVM FACULDADE INTEGRADA

PRÁTICA DE ENSINO DE ALGORITMO NO CURSO
UNIVERSITÁRIO DE CIÊNCIA DA COMPUTAÇÃO

Por: Emiliano Carlos Serpa Castor

Orientadora
Professora Edla Lucia Trocoli Xavier da Silva

Rio de Janeiro

2011

UNIVERSIDADE CANDIDO MENDES
PÓS-GRADUAÇÃO “LATO SENSU”
AVM FACULDADE INTEGRADA

PRÁTICA DE ENSINO DE ALGORITMO NO CURSO
UNIVERSITÁRIO DE CIÊNCIA DA COMPUTAÇÃO

Apresentação de monografia à Universidade
Candido Mendes como requisito parcial para
obtenção do grau de especialista em Docência do
Ensino Superior

Por: Emiliano Carlos Serpa Castor

AGRADECIMENTOS

À Edla Lucia Trocoli Xavier da Silva,
professora e orientadora, pela
colaboração.

Aos meus alunos da Universidade
Santa Úrsula, do Centro Universitário
Moacyr Sreder Bastos e da Faculdade
de Tecnologia SENAC Rio.

DEDICATÓRIA

Dedico este trabalho à minha família pelo amor, apoio, estímulo, paciência e colaboração.

RESUMO

A desenvoltura do aluno iniciante em um curso de Ciência da Computação em produzir algoritmos e, conseqüentemente, programas de computador na última década tem estado bem abaixo do esperado. A forma de conduzir um aluno, que traz consigo apenas as experiências prévias de mundo real, devem ser canalizadas ao pensamento estruturado, conforme conduz o capítulo I desta monografia.

O capítulo II procura estabelecer pontos de coerência da didática do ensino superior, conceitos de modelos de aprendizagem de Robert Gagné e aplicação em um curso de Ciência da Computação ou algum curso de ciências exatas que tenha no currículo algoritmos e programação de computadores.

O capítulo III recupera tópicos dos capítulos anteriores e busca a possibilidade de inserção desses pontos no currículo de ciência da computação, em termos de disciplinaridade e interdisciplinaridade.

Entende-se que este trabalho apresenta uma abordagem coerente que busca unir os conceitos pedagógicos às necessidades do estudante de computação, e das demais ciências exatas, na primeira década do século XXI.

METODOLOGIA

O método utilizado para o desenvolvimento do tema proposto baseou-se na leitura de livros, apostilas do curso de pós-graduação da UCAM / AVM internet e outras informações disponíveis na mídia, que tratam de práticas de ensino de algoritmos.

O desenvolvimento deste trabalho consistiu em reunir algumas fontes de informação como referência imediata para tratar o assunto com a devida importância que certamente será dada neste século XXI.

O propósito é reunir nesta monografia, informações muito importantes sobre prática de ensino de algoritmos. Tem por fim, transformá-la em uma fonte de consulta para aqueles que desejem iniciar estudos sobre técnicas de ensino de algoritmo para estudantes do ensino superior, pensamento cognitivo em forma de pensamento estruturado computacional e pensamento estruturado no currículo de ciência da computação.

SUMÁRIO

INTRODUÇÃO	08
CAPÍTULO I - Técnica de Ensino de Algoritmo para Estudantes do Ensino Superior	10
CAPÍTULO II - O Pensamento Cognitivo em forma de Pensamento Estruturado Computacional	15
CAPÍTULO III – O Pensamento Estruturado no Currículo de Ciência da Computação	28
CONCLUSÃO	36
BIBLIOGRAFIA CONSULTADA	38
ÍNDICE	40
FOLHA DE AVALIAÇÃO	41

INTRODUÇÃO

É possível desenvolver ou melhorar técnicas de ensino de algoritmos para alunos iniciantes em um curso universitário de Ciência da Computação?

É possível fazer com que o aluno consiga expressar rapidamente, após um período de adaptação e estudo, o seu pensamento cognitivo em forma de pensamento estruturado para um computador realizar uma tarefa?

O currículo de Ciência da Computação está organizado da melhor forma para ajudar o aluno a desenvolver o pensamento estruturado?

Uma grande questão nos cursos universitários de Ciência da Computação recai em qual linguagem computacional será introduzida ao aluno quando este cursar a primeira cadeira de programação de computadores. Diversas linguagens são defendidas pelas principais Instituições de Ensino, sempre calcadas em seus projetos pedagógicos, mas sempre recaem na dificuldade de estabelecer no aluno o pensamento estruturado e em fazê-lo de forma definitiva, ou seja, em garantir que o aluno vá aproveitar esses conhecimentos em estudos ou trabalhos futuros. A preocupação maior deveria recair em ajudar o aluno a desenvolver a sua cognição computacional de forma consistente.

Como objetivo geral, esse trabalho recai sobre a prática de ensino de algoritmo que deve ser iniciada com a desconexão da lógica humana como ligação com a máquina. O aluno precisa entender que a cognição do pensamento humano não pode ser inserida de imediato para comando de máquinas sem a devida adaptação. O aluno que entra na universidade e enfrenta pela primeira vez cadeira de programação de computadores tem o desafio de entender como se deve passar comandos ordenados logicamente para uma máquina. Essas técnicas precisam ser ensinadas em forma coordenada e cadenciada. A forma coordenada recai no currículo de Ciência da Computação, que precisa ter os pré-requisitos e co-requisitos definidos claramente. A forma cadenciada recai no tempo de adaptação e na forma como os alunos iniciantes serão conduzidos à produção de algoritmos computacionais. O Estudo se propõe abordar esses aspectos e apontar um

viés mais adequado para o tema, situando o aluno da primeira metade do século XXI como usuário habitual de softwares básicos e com a alta disseminação de informação que a Internet proporciona.

Como objetivo específico, este trabalho busca analisar o currículo básico de Ciência da Computação e Propor um viés mais adequado à prática de ensino de algoritmo ao aluno do século XXI.

A partir das escolas de Ensino Superior em Ciência da Computação do Estado do Rio de Janeiro, o trabalho busca validar as seguintes hipóteses:

- Duas disciplinas organizadas como co-requisitos, a saber: Algoritmos e Princípios de Programação, com 90 horas/aula por semestre cada uma, oferecidas no segundo período universitário e necessariamente após a disciplina de Introdução à Tecnologia da Informação. Este quadro deve formar um aluno com maior base computacional do que essas disciplinas de 60 horas e sem pré-requisitos como acontece na maioria das escolas superiores.
- As disciplinas de Algoritmos e Programação de Computadores atuando em conjunto de forma coordenada e cadenciada e, levando em conta as dificuldades inerentes e o aluno do século XXI, deve maximizar o desenvolvimento computacional cognitivo do aluno.

CAPÍTULO I

TÉCNICA DE ENSINO DE ALGORITMO PARA ESTUDANTES DO ENSINO SUPERIOR

...Deus é maior que todos os obstáculos.

É possível ter inúmeras definições para algoritmo. Uma delas, de Ascencio (1999), é a seguinte: “Algoritmo é a descrição de uma seqüência de passos que deve ser seguida para a realização de uma tarefa”. Outra, mais completa e estruturada diz que:

“Um algoritmo consiste simplesmente em uma seqüência finita de regras ou instruções que especificam como determinadas operações básicas, executáveis mecanicamente, devem ser combinadas para realização de uma tarefa desejada” (Camarão, 2003).

O final da primeira década do século XXI traz algumas percepções importantes no campo educacional. Dentre elas, percebe-se uma grande importância do profissional com habilidades e competências globais. Observa-se corriqueiramente que o domínio da microinformática é habilidade comum às profissões de nível superior no Brasil. Segmentando as carreiras, como o profissional de ciências exatas, percebe-se que o domínio de técnicas de inteligência computacional tornou-se mais importantes no dia a dia desse profissional. Arrisco a afirmar que os conceitos básicos de computação, como a compreensão de hardware e de técnicas de desenvolvimento de software, são os novos conhecimentos globais da área de tecnologia. O desenvolvimento da tecnologia requer pesquisas e projetos multidisciplinares com participações de profissionais de diferentes formações acadêmicas. Como seria uma comunicação entre pessoas de diferentes formações? Seria preciso ter um conhecimento comum mínimo para que as pesquisas e/ou projetos

pudessem ter um desenvolvimento adequado. O pensamento estruturado tem papel fundamental no conhecimento comum desses profissionais. Mas como desenvolvê-lo em um curso universitário na área de ciências exatas?

Uma grande questão nos cursos universitários de ciências exatas e, em particular, Ciência da Computação, recai em qual linguagem computacional será introduzida ao aluno quando este cursar a primeira cadeira de programação de computadores. Diversas linguagens são defendidas pelas principais Instituições de Ensino, sempre calcadas em seus projetos pedagógicos, mas sempre recaem na dificuldade de estabelecer no aluno o pensamento estruturado e em fazê-lo de forma definitiva que vá aproveitar em estudos ou trabalhos futuros. Nesse ponto, penso que a linguagem de programação C seja a mais adequada, principalmente nessas questões, as quais abordo ao decorrer desse trabalho. “A linguagem C foi criada por Dennis Ritchie, em 1972, no centro de Pesquisas da Bell Laboratories. Sua primeira utilização importante foi a reescrita do Sistema Operacional UNIX, que até então era escrito em assembly. Em meados de 1970 o UNIX saiu do laboratório para ser liberado para as universidades. Foi o suficiente para que o sucesso da linguagem atingisse proporções tais que, por volta de 1980, já existiam várias versões de compiladores C oferecidas por várias empresas, não sendo mais restritas apenas ao ambiente UNIX, porém compatíveis com vários outros sistemas operacionais. (...) C é uma linguagem de propósito geral, sendo adequada à programação estruturada. No entanto é mais utilizada escrever compiladores, analisadores léxicos, bancos de dados, editores de texto, etc. A linguagem C pertence a uma família de linguagens cujas características são: portabilidade, modularidade, compilação separada, recursos de baixo nível, geração de código eficiente, confiabilidade, regularidade, simplicidade e facilidade de uso.” (Celes, Waldemar; Ierusalimschy, Roberto, 2009).

Acredito que a linguagem C, assim como as demais linguagens de alto nível, seja bem aplicável aos estudos de programação de computadores pelo fato de cobrar do aluno um pensamento estruturado e com rigor que uma programação deve ter, o que no início do curso universitário é tipicamente

nebuloso pelo motivo de influência da lógica humana, e pelo fato de que é base das principais linguagens de mercado na presente década, como JAVA e C#. Coloco esse ponto em duas etapas.

A primeira refere-se à desconexão da lógica humana como ligação com a máquina. O aluno precisa entender que a cognição do pensamento humano não pode ser inserida de imediato para comando de máquinas sem a devida adaptação. Não me refiro a IA, inteligência artificial, que com o uso da lógica deôntica se torna mais próximo dos comandos humanos e sim do aluno que entra na universidade e enfrenta pela primeira vez cadeira de programação de computadores. Para ilustrar o caso, suponha que é preciso dizer a uma máquina que é necessária a troca de uma lâmpada queimada do teto da sala de estar de um apartamento. Usando a experiência prévia humana, o aluno provavelmente descreveria para a máquina os seguintes passos. É preciso ter uma lâmpada nova e uma escada e: (a) Suba na escada; (b) Troque a lâmpada. Percebe-se então que é necessário colocar a escada perto da lâmpada queimada. Falta a informação de onde se deve colocar a escada. Outra tentativa então: (a) Coloque a escada logo em baixo da lâmpada; (b) Suba na escada; (c) Troque a lâmpada. Percebe-se, dessa vez, que é necessário ter as mãos livres para trocar a lâmpada. A pessoa ficou sem mãos disponíveis para realizar a operação de troca. Mais uma tentativa: (a) Coloque a escada logo em baixo da lâmpada; (b) Deixe a lâmpada nova; (c) Suba na escada; (d) Retire a lâmpada nova do soquete; (e) Desça da escada; (f) Jogue fora a lâmpada velha; (g) Suba na escada levando a lâmpada nova; (h) Coloque a lâmpada nova no soquete; (i) Desça da escada. Agora sim a tarefa pode ser realizada com sucesso. É notório que trocar uma lâmpada é tão simples que qualquer um pode fazer sem precisar de roteiro. Qualquer uma pessoa, uma máquina não. Na elaboração de uma seqüência lógica, é preciso pensar em todos os detalhes da tarefa. É claro que um roteiro apresentado a um ser humano não precisa de tantos detalhes óbvios, pois temos a capacidade de raciocinar e decidir. Agora uma máquina executa apenas o que lhe é pedido. Para isso, as instruções precisam ser rigorosamente estruturadas, exatas e bem detalhadas. A esse roteiro para máquinas

chamamos de Algoritmo. É nesse ponto que os alunos precisam reconstruir o significado da cognição e aprender a forma de comunicação de linguagem de máquina, assim dizendo. Precisa-se gerar o caos e desmistificar a linguagem de máquina.

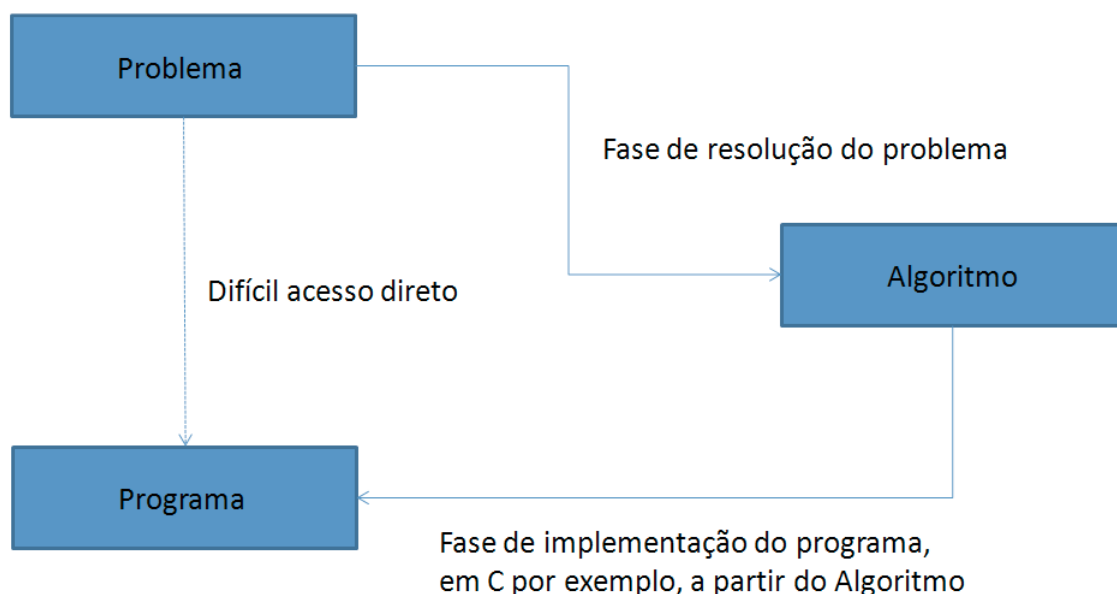


Figura 1.1 – Solução de problema com computador com uso de algoritmos (Lages, 1985).

Feito o algoritmo, o próximo passo é traduzi-lo em um programa de computador, usando uma linguagem de programação, que tem as suas próprias regras de sintaxe e semântica. As principais linguagens de mercado nas últimas décadas têm sido baseadas nas sintaxes da linguagem C, ou seja, suas estruturas e seus comandos são muito similares aos usados em C. Como típicos humanos, nos sentimos mais confortáveis em explorar algo que nos remeta a uma familiaridade do que enfrentar o novo e o completo desconhecido. A familiaridade com aspectos que já aprendemos anteriormente nos traz respostas frente aos estímulos de como fazer um determinado comando e obter soluções armazenadas em algum lugar no nosso cérebro, resultado este de uma interação do indivíduo com o ambiente computacional experimentada anteriormente, provavelmente no ambiente acadêmico.

Se é possível construir um pensamento estruturado em dizer a tarefa que a máquina tem que fazer e se já se domina em como dizer isso para a máquina, usando sintaxes lingüísticas da programação (em C por exemplo), tem-se o processo completo da produção de um programa de computador. Levando em conta a familiaridade da linguagem C, tipicamente acadêmica, com as demais linguagens de programação líderes de mercado, acredito que o estudo guiado em C, como aquisição de base sólida para o profissional de ciência da computação, seja imprescindível para o desenvolvimento do universitário.

CAPÍTULO II

O PENSAMENTO COGNITIVO EM FORMA DE PENSAMENTO ESTRUTURADO COMPUTACIONAL

“A arte de programar consiste na arte de organizar e dominar a complexidade.” (Dijkstra, 1998).

O psicólogo educacional americano Robert Mills Gagné (21 de agosto de 1916, 28 de abril de 2002) foi o primeiro a abordar pontos da ciência da instrução durante a Segunda Guerra Mundial para a força aérea. Posteriormente, Gagné passou a desenvolver uma série de estudos e trabalhos que ajudaram a codificar o que ele considerava como boa instrução. O modelo de ensino de aprendizagem proposto pelo Gagné mostra que a aprendizagem não se resume somente ao que ocorre de forma natural, como um ser humano aprende a andar, a respirar ou a falar. Seria também um acontecimento ocorrido sob certas condições que podem ser observadas. A respeito da teoria da aprendizagem e instrução, pode-se calcar no ensino superior em três principais aspectos:

- Taxonomia de Gagné
- Hierarquias de Habilidades Intelectuais
- Eventos de Instrução

2.1 – Taxonomias de Gagné

A Taxonomia de Gagné se resume aos efeitos de aprendizagem resultantes no processo de modificação na capacidade de compreensão do ser humano. Os passos desse processo são constantes interações do professor com os estudantes seguindo o que se aprende, situação estimuladora e resposta ou desempenho. A modificação atua na aprendizagem. Os estímulos

do ambiente fazem com que a informação seja processada e seja armazenada na memória do estudante, seguindo os passos de estímulo, processamento e resposta.

Ao iniciar um processo de aprendizagem em algoritmo o professor deve definir o que se deseja, como por exemplo, solução de problemas usando algoritmo.

O professor deve usar situações estimuladoras do cotidiano do aluno, lançando mão de seu conhecimento prévio de mundo. Como ilustração, suponha que após um banho, um homem deseja se vestir usando camisa, calça, cueca, meia, sapato, gravata, cinto e paletó. Para realizar essa tarefa, o estudante pode remeter-se às suas experiências prévias de vestir (ou calçar) cada uma das peças em uma seqüência que pode até variar, mas certamente terá que pôr as meias antes de calçar os sapatos, terá que vestir a cueca antes da calça, terá que colocar a gravata e o paletó após a camisa e terá que pôr o cinto após a calça. Tendo estes passos em mão o professor pode estimular os seus alunos a encontrarem formas válidas que respeitem as limitações estruturadas acima ou mesmo exemplificá-la em um primeiro momento e, em seguida, propor um novo problema do mesmo gênero. Como soluções válidas para essas tarefas como vestir a cueca, pôr as meias, vestir a camisa, vestir a calça, colocar os sapatos, colocar o cinto, colocar a gravata e vestir o paletó. Observe que vestir a camisa, colocar a gravata, vestir a cueca, vestir a calça, colocar o cinto, pôr as meias, calçar os sapatos e vestir o paletó também é uma opção válida. Ambas as soluções, ações seqüenciais, podem ser realizadas, em ordens diferentes, mas respeitando as condições impostas para o problema.

Como resposta ou desempenho dos estudantes, o professor deve observar a desenvoltura e a dificuldade geral e essa avaliação o fará a decidir em reforçar ainda mais com outros problemas do mesmo gênero ou passar para uma nova fundamentação de acordo com o programa da disciplina.

Passada a etapa de familiarização com algoritmo, o professor deve começar a abordar os pontos técnicos e frisar que o algoritmo necessita:

- Ser finito

- Ser detalhado em termos de ações bem definidas e não ambíguas
- Deve ser descrito ações que devem seguir uma sequência lógica e ordenada.

Tecnicamente os algoritmos podem ser representados nas seguintes formas:

- Em uma língua técnica, como inglês ou português, usualmente em português recebe o nome de Portugol (Lages, 1985). Nessa língua técnica são usados comandos no imperativo que indicam as ações a serem tomadas: IMPRIMA nome, LEIA valor, dentre outros (Lages, 1985).
- Em uma linguagem de programação, como Pascal, Cobol, C, C#, Java, dentre outros. É usada por programadores com mais experiência, pois apresenta certos inconvenientes como a utilização apenas das instruções existentes na linguagem e pelo fato de serem muito detalhadas nas sintaxes, o que desvia a atenção do programador da solução do problema em si.
- Representações gráficas, muito recomendáveis para o início do estudo pelo fato de sua representação ser feita por desenhos, o que facilita uma compreensão geral do problema (Lages, 1985). Destacam-se como representações gráficas: Fluxograma, Diagrama de Nassi-Scheniderman (DNS), Método de Jackson e Diagrama de Warnier-Orr (Lages, 1985).

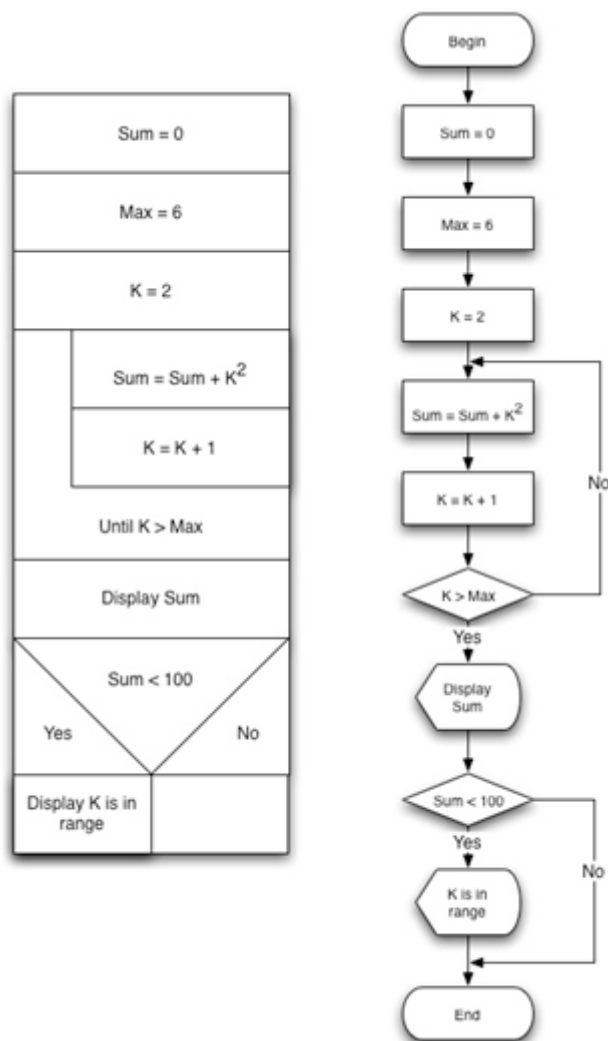


Figura 2.1 – Diagrama DNS e Fluxograma, respectivamente.

(<http://adrianhoe.com/adrianhoe/2008/09/12/nassi-shneiderman-diagram/>, 2011)

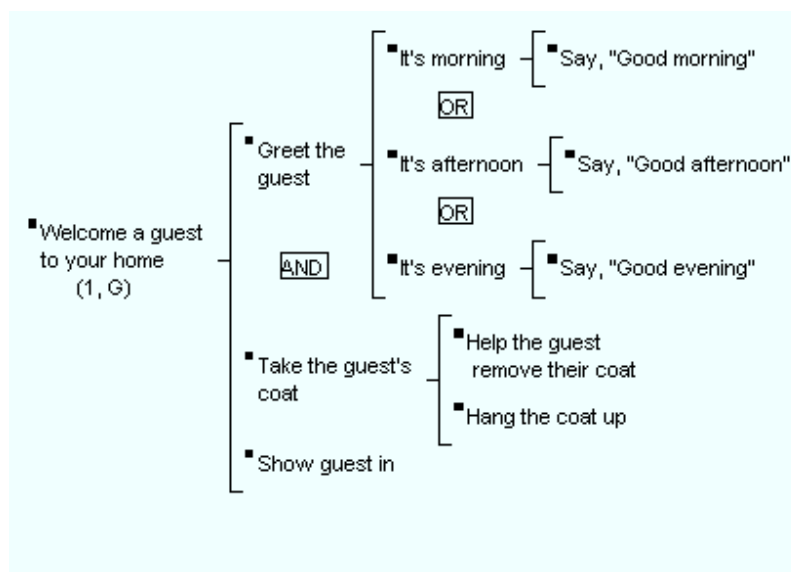


Figura 2.2 – Diagrama de Warnier-Orr

(http://varatek.com/warnierorr_diagrams.html, 2011)

Quando se deseja resolver um problema usando um computador, a sequência de passos que conduzem à solução é detalhada até que se chegue a um conjunto de ações primitivas que se denominam comandos ou instruções, e que podem ser entendidas e executadas pela máquina. O algoritmo, no maior nível de detalhamento possível, quando traduzido para uma linguagem de programação, constituirá um programa. Uma das vantagens de programas utilizando um algoritmo é que a partir dele o programador pode realizá-lo em qualquer linguagem de programação que conheça ou deseje. (Lages, 1985).

O professor pode fazer uso das representações gráficas como fonte motivadora para os alunos iniciantes, antes de abordar as demais formas de produção de algoritmos. De acordo com Gagné, o professor define o que se deseja, como por exemplo, solução de problemas usando algoritmo com uso de representações gráficas.

Em seguida, o professor define como situação estimuladora a solução de um problema como, por exemplo, a média aritmética de duas notas do aluno, ambas informadas pelo usuário. Caso a média tenha sido igual ou superior a seis, o aluno está aprovado. Caso contrário, reprovado. O professor

pode resolver a questão passo a passo e, em seguida, propor uma nova situação semelhante para os alunos e pedir-lhes que a solucionem.

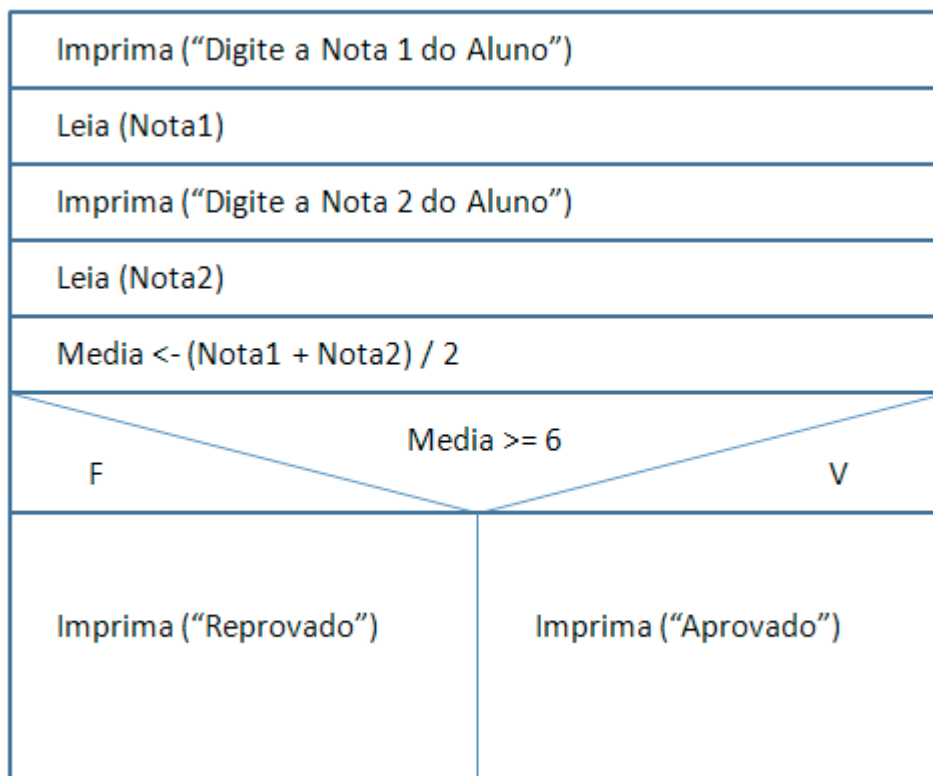


Figura 2.3 – Solução gráfica por DNS. Baseado em (Lages, 1985).

Novamente como resposta ou desempenho dos estudantes, o professor deve perceber a desenvoltura e a dificuldade geral e essa avaliação o fará a decidir em trabalhar ainda mais com outros problemas do mesmo gênero ou passar para uma nova fundamentação de acordo com o programa da disciplina.

Utilizar o português técnico seria a próxima etapa do processo de aprendizagem. Definição do que se deseja: solução de problemas usando algoritmo com português técnico.

O professor define, por exemplo, como situação estimuladora o mesmo problema sugerido na situação anterior: a da média aritmética.

ALGORITMO MÉDIA

```
// {DECLARAÇÃO DE VARIÁVEIS}
VAR
    REAL: Nota1, Nota2, Media
INICIO
    IMPRIMA ("Digite a Nota 1 do Aluno: ")
    LEIA (Nota1)
    IMPRIMA ("Digite a Nota 2 do Aluno: ")
    LEIA (Nota2)
    Media <- (Nota1 + Nota2) / 2
    SE (Media >= 6) ENTÃO
        IMPRIMA ("Aluno Aprovado")
    SENÃO
        IMPRIMA ("Aluno Reprovado")
    FIM SE
FIM
```

Figura 2.4 – Algoritmo representado em português técnico (Portugol).
Baseado em (Lages, 1985).

Mais uma vez, como resposta ou desempenho dos estudantes, o professor deve notar a forma como foi recebido esse desafio aos alunos e essa avaliação o fará decidir em trabalhar ainda mais com outros problemas do mesmo gênero ou passar para uma nova fundamentação seguindo o programa da disciplina.

A última etapa desse processo de aprendizagem seria trabalhar a codificação do algoritmo usando alguma linguagem de programação. Esse passo pode ser opcional, dependendo da ementa do programa. Como afirmado no capítulo 1 deste trabalho, a linguagem que mais se adéqua ao rigor estruturado de um programa de computação e também base para as linguagens de mercado da primeira década do século XXI, seria a linguagem

C. Definição do que se deseja: codificação de algoritmo a partir de seu texto em português técnico.

Para esta etapa do processo de aprendizagem ocorrer com mais tranqüilidade, as etapas anteriores de solução de problemas com algoritmo usando representação gráfica e com o uso de português técnico estejam assimilados pelos alunos de forma satisfatória e consistente.

A situação estimuladora, por exemplo, será codificar o algoritmo produzido em português técnico na etapa anterior. Usando a linguagem C/C++ a solução seria:

```
#include <stdio.h>
#include <conio.h>
int main( ) {
    //Declaração de Variáveis
    float nota1, nota2, media;
    //Início
    printf("\n Digite a Nota 1 do Aluno: ");
    scanf("%f", &nota1);
    printf("\n Digite a Nota 2 do Aluno: ");
    scanf("%f", &nota2);
    media = (nota1 + nota2) / 2;
    if(media >= 6) {
        printf("\n Aluno Aprovado");
    }
    Else {
        printf("\n Aluno Reprovado");
    }
    getch( );
    return(0);
}
```

Figura 2.5 – Programa derivado do algoritmo MÉDIA. Programa codificado na linguagem C. Baseado em (Lages, 1985).

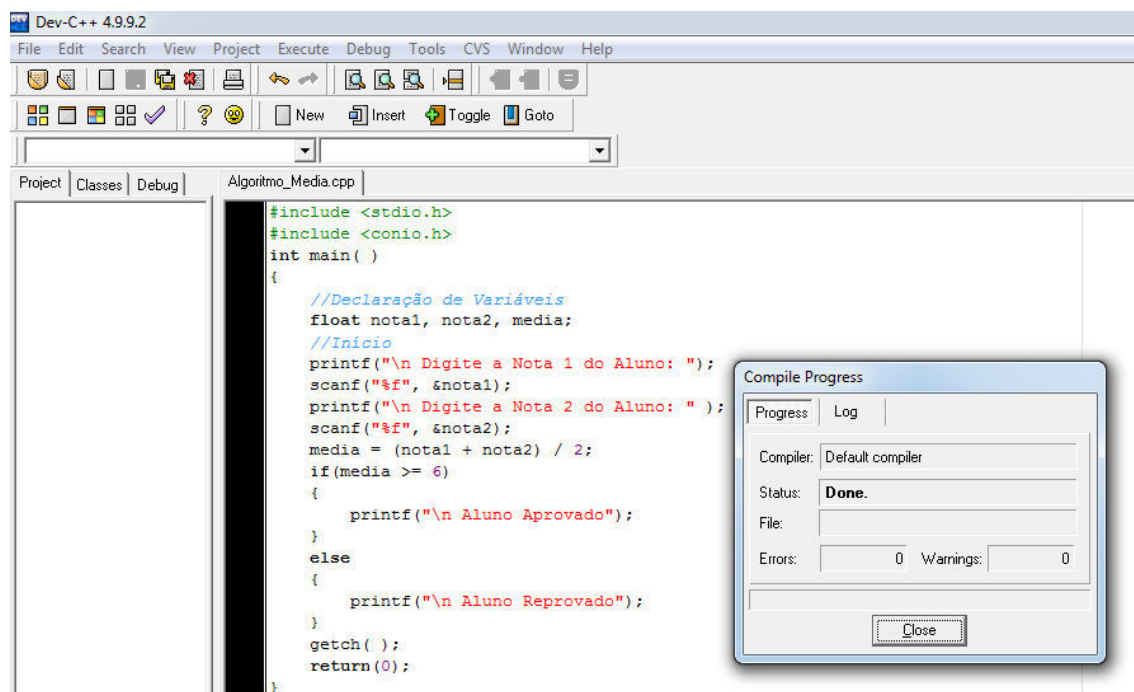


Figura 2.6 – Aviso de compilação do código na ferramenta DEV-C++. A compilação realiza a análise sintática e léxica do código e, se não houver erro, o programa estará pronto para ser executado. Baseado em (Lages, 1985).

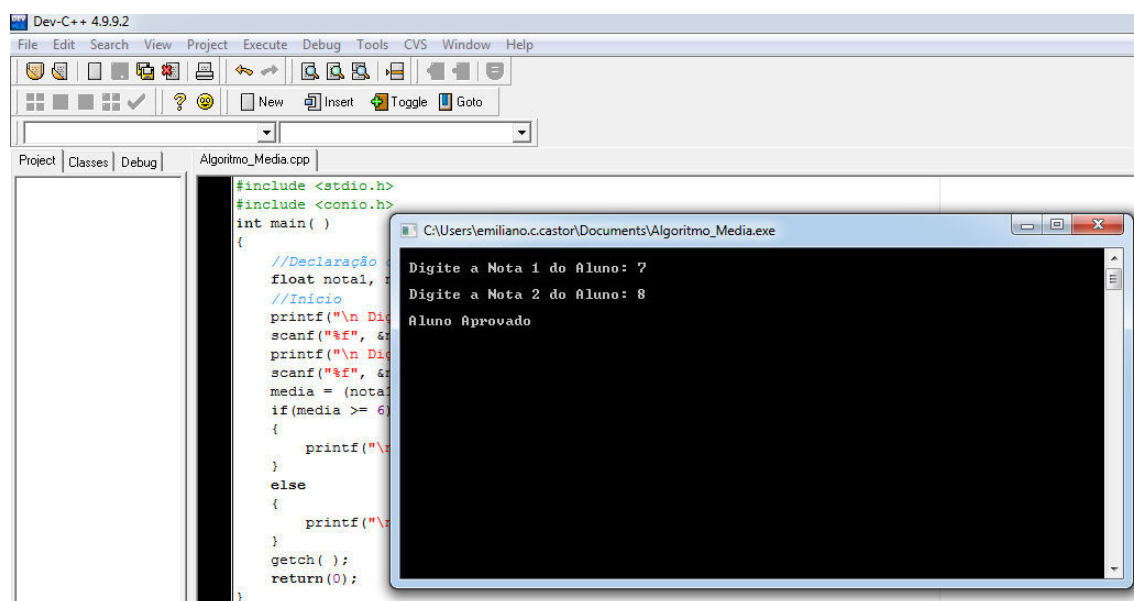


Figura 2.7 – Execução do programa derivado do algoritmo MÉDIA. Baseado em (Lages, 1985).

A partir dessa atividade, o professor deve fazer, se possível, a correspondência de cada linha de comando do código com as do texto em português técnico, enfatizando sempre que cada linguagem de programação tem a sua própria sintaxe. O texto em português técnico é geral, já a codificação é específica e varia entre cada linguagem de programação.

No exemplo de codificação, o que vem após as duas barras (//) são comentários e são ignorados pelo compilador da linguagem. Cada bloco precisa ter sua delimitação marcada, ou seja, onde começa e onde termina. Nesse exemplo percebem-se as delimitações do bloco principal do programa (*main*) e da estrutura de decisão (*if* e *else*). O comando *printf* refere-se ao IMPRIMA do texto em português técnico e o *scanf* ao LEIA. O leitor pode reparar que a sintaxe desses comandos é bem particular e está restrita à forma como a linguagem foi projetada. O comando de atribuição de valores (=) tem a correspondência ao <- do texto em português técnico. Por fim os comandos *getch()* segura a tela e espera por que qualquer tecla seja digitada para seguir a seqüência de comandos, e o *return (0)* é um artifício da linguagem que pede um retorno de um número inteiro que foi explicitada na linha *int main()*.

É comum que os estudantes apresentem dificuldades nas primeiras vezes que codifiquem um algoritmo. É preciso compreender bem a forma estruturada e simples de um algoritmo e também conhecer a sintaxe da linguagem de programação em uso. O professor deve estimular o constante uso de exercícios como forma de fazer o aluno refletir em como passar tais informações para o computador, de forma estruturada e organizada. O professor deve ressaltar a importância da programação nos dias atuais e estimulá-los a pesquisar os manuais técnicos da linguagem em como realizar os comandos que os textos em português técnicos prevêem.

2.2 – Hierarquias de Habilidades Intelectuais

São conjuntos de habilidades específicas que devem relacionar-se entre si a fim de gerar transferência positiva à aprendizagem de uma capacidade posterior. Como forma de organização a anterior é pré-requisito da

seguinte. As habilidades devem referir-se ao que o estudante é capaz de fazer e não necessariamente ao que ele já sabe fazer. São as seguintes habilidades de aprendizagem intelectuais, relacionadas com a construção do pensamento cognitivo na computação:

- Aprendizagem de Sinais – Condicionamento simples por meio de associação de estímulo e resposta. Deve ser trabalhada nas aulas de algoritmos estruturados.
- Aprendizagem de Condicionamento Operante – Estabelece associação entre estímulo, resposta e reforço. Deve ser trabalhada nas aulas de algoritmos estruturados.
- Aprendizagem de Cadeia Motora – Habilidade exigida nos esportes, nas artes, música, dentre outras. Sem relacionamento direto à aprendizagem de algoritmos computacionais.
- Aprendizagem de Associações Verbais – Encadeamentos de estímulo e resposta, presentes na aquisição da linguagem oral, da língua materna ou estrangeira. . Sem relacionamento direto à aprendizagem de algoritmos computacionais.
- Aprendizagem de Discriminações Múltiplas – Distinguir entre objetos semelhantes, perceber as diferenças e semelhanças e ter a capacidade de realizar comparações. Útil à generalização de situações de problemas de algoritmos. Deve ser trabalhada nas aulas de algoritmos estruturados.
- Aprendizagem de Conceitos – Compreensão dos conceitos concretos e conceitos definidos, sem os quais, o estudante permanece em um nível inferior de associações verbais. Deve ser trabalhada nas aulas de algoritmos estruturados, obedecendo as diferenças entre esses dois conceitos:
 - Conceito Concreto – Discriminações ou características relevantes ou não relevantes de um objeto ou situação.
 - Conceito Definido – Demonstração do significado de alguma classe de objetivos, de eventos ou de relações. Saber classificar.

As habilidades a seguir devem ser as mais trabalhadas no conceito computacional:

- Aprendizagem de Princípios – É a relação se-então, primordial para o conceito computacional. Para sua compreensão, o estudante precisa ter migrado do nível verbal para o nível da demonstração.
- Aprendizagem de Solução de Problemas – É a capacidade de resolver problemas usando o pensamento e reflexão. É o uso de experiências prévias semelhantes e, com o poder de generalização, ter a possibilidade de resolução das situações desafios ou problemas. Essa aprendizagem intelectual é de grande importância para o aluno. O professor deve estimular os estudantes à descoberta por meio de soluções semelhantes aos problemas, realizando perguntas, levantando dúvidas e, acima de tudo, conduzir o aluno a ter o *insight* e, como consequência, à descoberta.

2.3 – Eventos de Instrução

Para o ensino de algoritmos computacionais para estudantes iniciantes, são aplicáveis os seguintes eventos de instrução de Gagné pelo professor:

- a) Obter e dirigir a atenção do aluno: Controlar as situações estimuladoras e definir estratégias motivacionais
- b) Informar o objetivo da aprendizagem: O professor deve dizer o que se espera do aluno e tentar torná-lo cúmplice do processo de ensino. Como cúmplice, o estudante torna-se ciente de que tem uma meta a atingir.
- c) Estimular a evocação das habilidades: O professor deve evocar o que o aluno já sabe a partir do aprendizado anterior. Muito útil nos capítulos de matrizes unidimensionais e multidimensionais (vetores e matrizes) e capítulos de registros e arquivos.

- d) Apresentar o material-estímulo: O assunto deve ser mostrado de forma objetiva, sucinta e organizado.
- e) Promover a orientação da aprendizagem: O professor deve ter a consciência de que deve se adaptar às diferenças entre turmas e, dentro de cada turma, às diferenças individuais. Deve tentar guiar o estudante.
- f) Facilitar o desempenho: O professor deve realizar a demonstração de algoritmos semelhantes e encorajar o aprendiz a desenvolver um válido.
- g) Realizar o *feedback* do desempenho certo: É muito importante a avaliação o mais próximo possível ao exercício. O professor deve mostrar os acertos com o intuito de levar o aluno a ter uma noção de sua evolução ao longo do curso.
- h) Avaliar o desempenho: O professor passa um juízo por meio de lista de exercícios, testes e pela evolução dos *feedbacks* que deu ao aluno ao longo do curso. Esses itens são importantes para o aluno de computação, pois esses passos realizam a retenção do conceito no estudante.
- i) Aumentar a retenção e transferência: A prática de algoritmos, com muitos exercícios e desafios é fundamental para internalização do conceito estruturado computacional.

O professor pode guiar a aquisição do pensamento estruturado computacional pelo quadro abaixo:

Etapas de Aprendizagem	Eventos
Pré-condições	(a), (b), (c)
Aprendizagem Plena	(d), (e), (f)
Retenção e Transferência	(g), (h), (i)

Quadro 1 – Pensamento Estruturado computacional com os Eventos de Instrução de Gagné (Gagné, 1968).

CAPÍTULO III

O PENSAMENTO ESTRUTURADO NO CURRÍCULO DE CIÊNCIA DA COMPUTAÇÃO

“Programar é como andar de bicicleta. Não se aprende vendo os outros fazendo. Depois que se aprende não mais se esquece.” (Autor desconhecido).

Os capítulos I e II deste trabalho abordam respectivamente como envolver o aluno recém chegado ao ensino superior com algoritmos computacionais e como desenvolver nesse estudante o pensamento estruturado segundo conceitos de Robert Gagné. Tão importante quanto essas questões está a discussão do ensino de algoritmo e programação no currículo do ensino superior na área de exatas, em particular no curso de Ciência da Computação.

As escolas de ensino superior em Ciência da Computação do estado do Rio de Janeiro apresentam um currículo variado. Muitas dessas escolas, na sua maioria privada, têm optado pela simplificação do currículo, reduzindo drasticamente a carga horária do curso a ponto de trocarem, inclusive, a denominação do curso. No lugar de Ciência da Computação, com disciplinas de todas as vertentes da computação (hardware, programação, softwares comerciais, softwares científicos, compiladores, redes, dentre outros), tem-se tornado mais freqüente a denominação de Sistemas de Informação, que não necessariamente apresenta todas as vertentes da computação no currículo e, as que apresentam, tem na ementa grande redução de carga horária e conteúdo em relação às disciplinas correspondentes no curso de Ciência da Computação. Essa redução é preocupante no ponto de vista do desenvolvimento pedagógico das disciplinas programação já que são de fundamental importância para o profissional de tecnologia da informação.

Consulta Interativa Consulta Textual Consulta Avançada

UF: RJ CURSO: CIÊNCIA DA COMPUTAÇÃO MUNICÍPIO: SELECIONE UM MUNICÍPIO Presencial Distância Fechar

Cursos (480)

- BIOTECNOLOGIA
- CIÊNCIA AMBIENTAL
- CIÊNCIA DA COMPUTAÇÃO**
- CIÊNCIA DA EDUCAÇÃO
- CIÊNCIA ECONÔMICA
- CIÊNCIA E TECNOLOGIA DE ALIMENTOS
- CIÊNCIA POLÍTICA
- CIÊNCIAS AERONÁUTICAS - AERONAVES DE ASA FIXA

Municípios (58)

- ANGRA DOS REIS
- ARARUAMA
- ARRAIAL DO CABO
- BARRA DO PIRAÍ
- BARRA MANSA
- BELFORD ROXO
- BOM JESUS DO ITABAPOANA
- CABO FRIO
- CAMARUÍ

Instituições (19)

- CENTRO UNIVERSITÁRIO MUATYK SREDER BASTOS - UNIMSB
- CENTRO UNIVERSITÁRIO PLÍNIO LEITE - UNIPLI
- CENTRO UNIVERSITÁRIO SERRA DOS ÓRGÃOS - UNIFESO
- PONTIFÍCIA UNIVERSIDADE CATÓLICA DO RIO DE JANEIRO - PUC-RIO
- UNIVERSIDADE CÂNDIDO MENDES - UCAM
- UNIVERSIDADE CATÓLICA DE PETRÓPOLIS - UCP
- UNIVERSIDADE DO ESTADO DO RIO DE JANEIRO - UERJ
- UNIVERSIDADE ESTADUAL DO NORTE FLUMINENSE DARCY RIBEIRO - UENF
- UNIVERSIDADE FEDERAL DO RIO DE JANEIRO - UFRJ

Figura 3.1 – Listagem das Instituições de Ensino Superior com curso de Ciência da Computação, segundo o site e-MEC: 19 cursos (<http://emec.mec.gov.br/> , 2011)

Consulta Interativa Consulta Textual Consulta Avançada

UF: RJ CURSO: SISTEMAS DE INFORMAÇÃO MUNICÍPIO: SELECIONE UM MUNICÍPIO Presencial Distância Fechar

Cursos (480)

- SERVIÇO SOCIAL
- SERVIÇOS PENAS
- SISTEMA DE INFORMAÇÃO
- SISTEMAS DE COMPUTAÇÃO
- SISTEMAS DE INFORMAÇÃO**
- SISTEMAS DE TELECOMUNICAÇÕES
- SISTEMAS ELÉTRICOS
- SISTEMAS PARA INTERNET
- SOCIOLOGIA

Municípios (58)

- ANGRA DOS REIS
- ARARUAMA
- ARRAIAL DO CABO
- BARRA DO PIRAÍ
- BARRA MANSA
- BELFORD ROXO
- BOM JESUS DO ITABAPOANA
- CABO FRIO
- CAMARUÍ

Instituições (23)

- CENTRO UNIVERSITÁRIO DE VOLTA REDONDA - UNIFOA
- CENTRO UNIVERSITÁRIO GERALDO DI BIASE - UGB
- FACULDADE DE INFORMÁTICA LEMOS DE CASTRO - FILC
- FACULDADE DE REABILITAÇÃO DA ASCE - FRASCE
- FACULDADE MORAES JÚNIOR - MACKENZIE RIO - FMJ - MACKENZIE RIO
- FACULDADE PARAÍSO - FAP
- FACULDADES INTEGRADAS CAMPO-GRANDENSES - FIC
- FACULDADES INTEGRADAS DE JACAREPAQUÁ - FIJ

Figura 3.2 – Listagem das Instituições de Ensino Superior com curso de Sistemas de Informação, segundo o site e-MEC: 23 cursos (<http://emec.mec.gov.br/> , 2011)

UF: RJ CURSO: SISTEMA DE INFORMAÇÃO MUNICÍPIO: SELECIONE UM MUNICÍPIO Presencial Distância Fechar

Cursos (480)

- SERVIÇO SOCIAL
- SERVIÇOS PENAIS
- SISTEMA DE INFORMAÇÃO**
- SISTEMAS DE COMPUTAÇÃO
- SISTEMAS DE INFORMAÇÃO
- SISTEMAS DE TELECOMUNICAÇÕES
- SISTEMAS ELÉTRICOS
- SISTEMAS PARA INTERNET
- SOCIOLOGIA

Municípios (58)

- ARARUAMA
- ARRAIAL DO CABO
- BARRA DO PIRAI
- BARRA MANSA
- BELFORD ROXO
- BOM JESUS DO ITABAPOANA
- CABO FRIO
- CAMBUCI

Instituições (16)

- ABEU - CENTRO UNIVERSITÁRIO - UNIABEU
- CENTRO UNIVERSITÁRIO DA CIDADE - UNIVERCIDADE
- CENTRO UNIVERSITÁRIO MOACYR SREDER BASTOS - UNIMSB
- FACULDADE DE CIÊNCIAS ECONÔMICAS, ADMINISTRATIVAS E DA COMPUTAÇÃO DOM BOSCO - FCEACDB
- FACULDADE DE FILOSOFIA, CIÊNCIAS E LETRAS DE DUQUE DE CAXIAS - FFCLDC
- FACULDADE DE TECNOLOGIA INFNET RIO DE JANEIRO - INSTITUTO INFNET
- FACULDADE FLAMA
- FACULDADE GAMA E SOUZA - FGS

Figura 3.3 – Listagem das Instituições de Ensino Superior com curso de Sistema de Informação, segundo o site e-MEC: 16 cursos (<http://emec.mec.gov.br/>, 2011). Note que a denominação desse curso para a da figura anterior é a mesma semanticamente.

UF: RJ CURSO: SISTEMAS DE COMPUTAÇÃO MUNICÍPIO: SELECIONE UM MUNICÍPIO Presencial Distância Fechar

Cursos (480)

- SERVIÇO SOCIAL
- SERVIÇOS PENAIS
- SISTEMA DE INFORMAÇÃO
- SISTEMAS DE COMPUTAÇÃO**
- SISTEMAS DE INFORMAÇÃO
- SISTEMAS DE TELECOMUNICAÇÕES
- SISTEMAS ELÉTRICOS
- SISTEMAS PARA INTERNET
- SOCIOLOGIA

Municípios (58)

- ARARUAMA
- ARRAIAL DO CABO
- BARRA DO PIRAI
- BARRA MANSA
- BELFORD ROXO
- BOM JESUS DO ITABAPOANA
- CABO FRIO
- CAMBUCI

Instituições (1)

- UNIVERSIDADE FEDERAL FLUMINENSE - UFF

Figura 3.4 – Listagem das Instituições de Ensino Superior com curso de Sistemas de Computação, segundo o site e-MEC: 1 curso (<http://emec.mec.gov.br/>, 2011). Note que a denominação desse curso para as das duas figuras anteriores é a mesma semanticamente.

UF: RJ CURSO: INFORMÁTICA MUNICÍPIO: SELECIONE UM MUNICÍPIO Presencial Distância Fechar

Cursos (480)

- HISTÓRIA - LICENCIATURA OU BACHARELADO
- HORTICULTURA
- HOTELARIA
- INFORMÁTICA**
- INTERDISCIPLINAR EM EDUCAÇÃO NO CAMPO
- INTERIORES E DECORAÇÕES
- INTERNET E REDES DE COMPUTADORES
- INTERPRETAÇÃO DA LÍNGUA BRASILEIRA DE SINAIS

Municípios (58)

- ARARUAMA
- ARRAIAL DO CABO
- BARRA DO PIRAÍ
- BARRA MANSÁ
- BELFORD ROXO
- BOM JESUS DO ITABAPOANA
- CABO FRIO
- CAMBUCI

Instituições (3)

- FACULDADES INTEGRADAS CAMPO-GRANDENSES - FIC
- UNIVERSIDADE DO ESTADO DO RIO DE JANEIRO - UERJ
- UNIVERSIDADE DO GRANDE RIO PROFESSOR JOSÉ DE SOUZA HERDY - UNIGRANRIO

Figura 3.5 – Listagem das Instituições de Ensino Superior com curso de Informática, segundo o site e-MEC: 3 cursos (<http://emec.mec.gov.br/> , 2011). Note que a denominação desse curso difere dos anteriores.

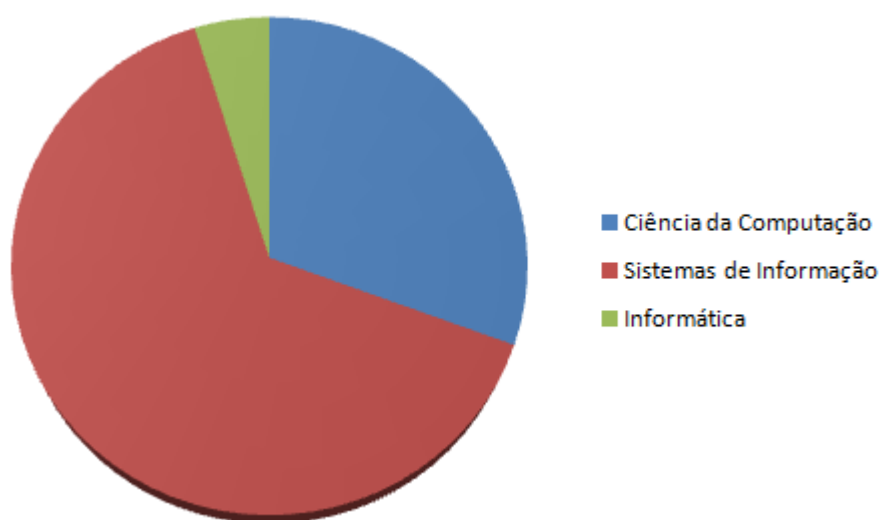


Figura 3.6 – Gráfico elaborado na ferramenta MS-Excel com a distribuição dos cursos de Ciência da Computação, Sistemas de Informação e Informática, no Estado do Rio de Janeiro. Baseado em (<http://emec.mec.gov.br/> , 2011).

As figuras 3.1 a 3.6 evidenciam a tendência de compactação do curso de Ciência da Computação. Até o mês de agosto de 2011, percebe-se que cerca de 65% dos cursos cadastrados entre esses três tipos de carreiras são os de Sistemas de Informação, mais do que o dobro em relação a Ciência da Computação, que registrou apenas cerca de 30% dos cursos cadastrados. O curso de Informática somou ao redor de 5% das opções. Para essas estatísticas foram considerados como o mesmo curso acadêmico as carreiras de Sistemas de Informação, Sistema de Informação (a palavra Sistema no singular) e Sistemas de Computação. O cadastro do MEC (Ministério da Educação) ainda relaciona outros tipos de cursos em relação à Tecnologia da Informação, mas estes fogem ao escopo por se tratarem de cursos de Engenharia ou de graduação tecnológica. Essa pesquisa foi feita para cursos de bacharelado de 3,5 a 4 anos, ou seja, graduação tradicional na área de computação.

De acordo com o apresentado nos capítulos anteriores, a abordagem ao aluno iniciante e as formas de representação do algoritmo, como gráfica e de texto em português técnico, cabem perfeitamente em uma disciplina com uma denominação genérica de Algoritmos. Essa disciplina pode ter na ementa os seguintes pontos, divididos em unidades:

Unidade I:

- ☐ Lógica;
- ☐ Algoritmo - Geral;

Unidade II:

- ☐ Comandos de entrada/saída;
- ☐ Algoritmo com Representação Gráfica;
- ☐ Algoritmo em Portugol;

Unidade III:

- ☐ Funções;
- ☐ Estruturas condicionais;
- ☐ Estruturas repetitivas;
- ☐ Recursividade;

A carga horária para essa disciplina deveria ser de 90 horas, que teria como distribuição sugerida 10 horas para Unidade I, 30 horas para Unidade II e 50 horas para Unidade III. Dessa forma seria possível trabalhar de forma consistente a base com a Unidade I (tomando por base o capítulo I desta monografia), aprofundar os conceitos de algoritmos (tomando por base o capítulo II desta monografia) e, finalmente, com a Unidade III, trabalhar em conjunto e em paralelo com a disciplina de programação de computadores, levando em conta, mais uma vez, os conceitos do capítulo II deste trabalho.

A disciplina de Programação de Computadores deveria ser oferecida paralelamente à disciplina de algoritmos e, de preferência, com o mesmo professor. Nesse ponto haveria uma facilidade de adequação de ambas as ementas, o que, teoricamente, ajudaria na unidade e no desenvolvimento de ambos os programas. Essa disciplina deveria ser, em sua quase totalidade, de caráter prático, ou seja, com utilização de laboratórios de informática, devidamente equipado com computadores configurados de acordo com o contexto atual de maquinaria.

Como abordado no capítulo I, a linguagem a ser utilizada na disciplina de programação deveria ser o C (ou mesmo o C++, que poderia ser utilizado em outras disciplinas de programação mais avançadas), devido ao seu rigor na sintaxe e na estruturação dos programas, além de ser base das linguagens mais comerciais da atualidade.

A disciplina de Programação de Computadores pode ter na ementa os seguintes pontos, divididos em unidades:

Unidade I:

A estruturação da linguagem C

A linguagem C e Tipos de Informação

A linguagem C e Tomada de decisões

Unidade II:

A linguagem C e *loopings*

A linguagem C Tabelas em Memória

Unidade III:

A linguagem C e suas funções e suas bibliotecas

Unidade IV:

A linguagem C e Arquivos

A linguagem C e paradigmas da programação

A carga horária para essa disciplina deveria ser de 60 horas, que teria como distribuição sugerida 15 horas para cada unidade. Essa disciplina pode basear-se nos conceitos do capítulo II desta monografia e deveria ser iniciada juntamente com a Unidade II da disciplina de Algoritmos. Dessa forma, o aluno já poderia ter internalizado os conceitos computacionais de algoritmos (capítulo I deste trabalho), que já teria sido abordado na Unidade I dessa disciplina.

É de extrema importância que o professor, a princípio o mesmo para as duas disciplinas, cadencie o programa de forma que os conceitos correspondentes de ambas as disciplinas caminhem juntos e de forma harmônica.

A Unidade I da disciplina de Algoritmos deveria caminhar sozinha, sem ainda o início da disciplina de programação. Isso concederá tempo para assentamento dos conceitos computacionais e à quebra de paradigma do cotidiano do estudante para a programação estruturada para máquinas.

O início da disciplina de Programação deveria iniciar juntamente com a Unidade II de Algoritmos e o professor deve usar o máximo de interdisciplinaridade entre essas duas cadeiras.

Como forma de avaliação, o ideal seria compor uma lista de exercícios em que os alunos deveriam desenvolver os algoritmos ora na forma gráfica, ora na forma textual para a cadeira de Algoritmos e os mesmos exercícios desenvolvidos na linguagem C para a disciplina de Programação de Computadores. Isso garante a interdisciplinaridade. Uma prova é uma opção a ser avaliada pelo professor, dependendo da situação e dos estudantes. Porém, vale salientar que uma prova deixa os alunos emocionalmente mais vulneráveis e por esse motivo, vale a pena em pensar em listas de exercícios mais complexas e uma prova mais objetiva, levando em conta que o aluno tem

mais recursos e tempo disponíveis para pesquisa em sua residência/universidade, em face de apenas algumas horas para realização de um exame (escrito ou prático).

O importante deste capítulo é a forma integrada e absolutamente coerente que as disciplinas de Algoritmos e Programação de Computadores devem ser conduzidas. Este fato, respeitando as sugestões de ementa e carga horária, juntamente com os conceitos pedagógicos adaptados ao aluno de ciências exatas mostradas neste trabalho, pode resultar em um desenvolvimento cognitivo computacional mais adequado a um estudante do século XXI.

CONCLUSÃO

Como se pode verificar nos conceitos de modelo de aprendizagem de Robert Gagné, a determinação dos objetivos equivale a capacidades a serem dominadas. A pedagogia ensina a planejar e a antecipar problemas de adaptação do estudante, da carga horária, das ementas, dentre outras.

Em particular as disciplinas de programação e/ou algoritmos no curso de Ciência da Computação, ou outra carreira que tenha essas disciplinas no currículo, estão tão reduzidas que não se tem como determinar objetivos viáveis. Procurei neste trabalho unir a necessidade de um desenvolvimento mais adequado do corpo discente, com uma pedagogia mais apropriada e com disciplinas mais coerentes.

Atualmente a maneira como a disciplina de programação e/ou algoritmos é conduzida evidencia um descompasso com todas as inovações e técnicas educacionais mais atualizadas, muitas vezes por ser uma disciplina da área técnica e o professor, em muitos casos, não ter uma formação pedagógica adequada, o que deixa ainda mais complicada a aprendizagem dos alunos. Este trabalho procurou elencar opções frente a um ensino tradicional, que vem sendo desenvolvido nas escolas de ensino superior há pelo menos duas décadas. Vale colocar que a demanda por dedicação e tempo do professor será mais exigida. O professor deverá procurar meios de abordar as hierarquias de aprendizagem de Gagné na turma que estiver responsável, sempre individualizando os métodos turma por turma e, por fim, estudante a estudante.

Claro que as teorias de Gagné não são respostas definitivas para a prática de ensino de algoritmos na universidade. São opções que aproveitam as condições humanas e tentam, no devido momento, sugar todas as possibilidades de compreensão de um tema tão complexo quando visto pela primeira vez.

O professor deve lembrar-se que o objetivo da didática é orientar a aprendizagem dos alunos para que realizem seus próprios passos, de modo que consigam traduzi-la em crescimento pessoal e em algo significativo para

as suas vidas. É dever do professor enxergar a didática não como a simples superposição de conteúdos, e sim como uma forma integrada e contextualizada de conhecimentos, que utiliza da melhor maneira a maturidade mental, emocional e social de cada estudante.

É verdade que a desistência ou reprovação de grande parte dos alunos nessas disciplinas está associada a uma falta de preparo dos próprios para a vida no ensino superior. Isso ocorre, muitas vezes, em algumas instituições particulares de ensino, onde o processo seletivo é menos concorrido e acabam ingressando alunos com pouca base, em sua maior parte com deficiência nos conteúdos de matemática, disciplina tão importante para o entendimento de algoritmos.

As hipóteses apresentadas no início deste trabalho guiam para um desenvolvimento mais sustentável do aluno de base mediana, e que devem ser corroboradas pelas Instituições de Ensino Superior, pois envolve aumento de carga horária, tendência esta inversa ao que ocorre em pleno início da segunda década do século XXI.

Eis um desafio que fica para os educadores, mantenedoras das Instituições de Ensino Superior e coordenadores de curso, afinados com a construção de uma perspectiva emancipatória de algoritmos e programação de computadores.

BIBLIOGRAFIA CONSULTADA

[Dahl, Dijkstra, Hoare 72] O-J Dahl, E. W. Dijkstra, C. A. R. Hoare, Structured Programming, Academic Press, London, 1972.

[Dijkstra 72A] E. W. Dijkstra, Notes on Structured Programming in [Dahl, Dijkstra, Hoare 72]

[Dijkstra 72B] E. W. Dijkstra, The Humble Programmer, Comm. of the ACM, Vol 15, 10, pp. 859-866, Outubro 1972.

[Jensen, Wirth 88] K. Jensen e N. Wirth, Pascal User Manual and Report, Springer, 1988.

[Lippman 91] S. B. Lippman, C++ Primer, Addison-Wesley, Reading Massachusetts, 1991.

[Parnas 72] D.L. Parnas. On the Criteria to be Used in Decomposing Systems into Modules. Comm. of the ACM, vol 15, 12, pp. 1053-1058, Dezembro, 1972.

[Ross, Goodenough, Irvine 1975], D. Ross, J. Goodenough e C. Irvine, Software Engineering: Processes, Principles, and Goals, IEEE Computer, Maio 1975.

[Stroustrup 91] B. Stroustrup, The C++ Programming Language, Addison-Wesley, Reading Massachusetts, 1991.10

[Lages, 1985] Lages, Newton Alberto de Castilho; Guimarães, Angelo de Moura, Algoritmos e Estrutura de Dados, LTC - Livros Técnicos e Científicos Editora S.A.

[Ascencio, 1999] Ascencio, Ana Fernanda Gomes. Lógica de programação com pascal. São Paulo: Makron Books, 1999.

[Camarão, 2003] Camarão, Carlos; Figueiredo, Lucília, Programação de Computadores em Java, LTC, 2003.

[Gagné, 1968] Gagné, R. M., Essentials of learning for instruction. 2. ed. Englewood, Cliff, NJ : Prentice-Hall, 1968.

[Ierusalimschy, Celes, 2009] Ierusalimschy, Roberto; Celes, Waldemar, Apostila de Programação, 2009.

http://varatek.com/warnierorr_diagrams.html, endereço acessado em 06/07/2011

<http://adrianhoe.com/adrianhoe/2008/09/12/nassi-shneiderman-diagram/>, endereço acessado em 06/07/2011

<http://emec.mec.gov.br/>, endereço acessado em 05/08/2011

ÍNDICE

FOLHA DE ROSTO	02
AGRADECIMENTO	03
DEDICATÓRIA	04
RESUMO	05
METODOLOGIA	06
SUMÁRIO	07
INTRODUÇÃO	08
 CAPÍTULO I	
Técnica de Ensino de Algoritmo para Estudantes Ens.Sup.	10
 CAPÍTULO II	
O Pensamento Cognitivo em forma de Pensamento Estruturado Computacional.	15
2.1 - Taxonomia de Gagné	15
2.2 – Hierarquias de Habilidades Intelectuais	24
2.3 – Eventos de Instrução	26
 CAPÍTULO III	
O Pensamento Estr. no Currículo de Ciência da Comp.	28
 CONCLUSÃO	36
BIBLIOGRAFIA CONSULTADA	38
ÍNDICE	40

FOLHA DE AVALIAÇÃO

Nome da Instituição:

Título da Monografia:

Autor:

Data da entrega:

Avaliado por:

Conceito: