ENSINO DE COMPUTAÇÃO PARA ESTUDANTES DE ENGENHARIA GCETE'2005

Edson de A. R. Barros [1], Lincoln C. Zamboni [2] and Sérgio V. D. Pamboukian [3]

Abstract *— The students of engineering accumulate, during their time in university, knowledge of a lot of technical competences, including how to develop scientific programs. The hole of training computation is more than explain the recent evolution technologists. Training computation is important in the process of formation the ratiocination logical-mathematician which is very important for the engineering works. Some professors, who work in Universidade Presbiteriana Mackenzie, during more than ten years, develop a methodology to teach computation. The methodology guaranty that the student's attention with a pleasure visual way of programming. The methodology doesn't forget the classic representation of logics, such as flowcharts and diagrams. The work is the base of three books: "Delphi para Universitários", "C++ Builder para Universitários" and "Cálculo Numérico para Universitários".*

Index Terms — *Methodology of teaching, Informatics, Programming, C++ Builder, Delphi.*

INTRODUÇÃO

No curso de engenharia existem várias disciplinas que evoluíram de acordo com o progresso da tecnologia. Lecionar tais assuntos exige dos professores uma relativa capacidade de adaptação e criatividade, bem como uma predisposição a uma contínua reciclagem.

O objetivo do presente texto é relatar um pouco da experiência adquirida por uma equipe de professores que se dedicaram às disciplinas de Computação na Escola de Engenharia da Universidade Presbiteriana Mackenzie (EEUPM).

DISCIPLINA DE COMPUTAÇÃO

Nos últimos 20 anos as disciplinas de Computação passaram por uma série de mudanças significativas que variaram desde os equipamentos e os ambientes operacionais até mesmo as próprias linguagens de programação, tenham sido elas estruturadas ou baseadas em conceitos de orientação a objetos.

Várias modificações de currículo foram propostas neste período, contudo uma linha mestra aparenta orientar as aulas e trabalhos sobre o tema. A própria literatura técnica disponível nas livrarias sofreu tantas mudanças a ponto de se tornar praticamente inadequada ao ensino, pois eram elaboradas segundo as necessidades comerciais das editoras e dos detentores das marcas de hardware e softwares disponíveis no mercado.

Mesmo com todo esse cenário acadêmico em constante mutação a Computação continuou sendo importante para o curso de engenharia. E é basicamente essa relevância que se pretende demonstrar.

Seu papel no curso de engenharia

Há algumas gerações os estudantes de engenharia usavam como ferramenta de trabalho a antiga régua de cálculo. Os livros eram engrossados por tabelas logarítmicas e trigonométricas e a forma de pensar nos problemas que envolviam cálculos complexos era moldada em uma certa lógica, a lógica de cálculo do engenheiro. A mesma lógica que começou a ficar evidente com o aparecimento das primeiras calculadoras eletrônicas e dos antigos e robustos computadores.

O aluno de então era oriundo de um ensino médio reforçado onde provavelmente tinha aprendido a falar latim e sem dúvida tinha estudado filosofia. De certa forma já estava sendo doutrinado a desenvolver bem um raciocínio lógico-matemático.

Nas últimas duas décadas, porém, um novo ferramental de cálculo começou a equipar os estudantes de engenharia. A régua de cálculo tornou-se objeto extremamente obsoleto e a informática, através das calculadoras e computadores, evoluiu de maneira tão assustadora que até os profissionais mais dedicados possuíam dificuldades em acompanhar a seu progresso. Os alunos vinham de escolas que os preparavam para um vestibular extremamente competitivo, onde não havia espaço para disciplinas não práticas. Os calouros passam a adquirir calculadoras científicas sofisticadíssimas e começam a possuir em casa um computador pessoal como um eletrodoméstico.

Em algum momento, em um passado não muito distante, uma disciplina destinada a ensinar como

[1] Edson de Almeida Rego Barros , professor da Escola de Engenharia da Universidade Presbiteriana Mackenzie, Rua da Consolação, 930, 01302-907, São Paulo, SP, Brazil, prof_edson@mackenzie.com.br
[2] Lincoln César Zamboni , professor da Escola de Engenharia da Universidade Presbiteriana Mackenzie, Rua da Consolação, 930, 01302-907, São Paulo, SP, Brazil, Lincoln.zamboni@mackenzie.com.br
[3] Sérgio Vicente Denser Pamboukian , professor da Escola de Engenharia da Universidade Presbiteriana Mackenzie, Rua da Consolação, 930, 01302-907, São Paulo, SP, Brazil, sergiop@mackenzie.com.br

transformar um computador em uma ferramenta de trabalho do profissional se fez necessária no curso de engenharia.

Com a rápida evolução dos equipamentos e das linguagens de programação algo começou a ficar evidente. O propósito da disciplina não era ensinar a lidar com uma máquina específica, um modelo de um fabricante qualquer, ou mesmo algum software determinado. O propósito da disciplina é auxiliar o estudante de engenharia a organizar suas idéias de uma maneira racional e clara, de tal forma que seja fácil programar em uma máquina, uma calculadora ou um computador, a seqüência de operações ou cálculo que se necessita fazer.

Na verdade o que se ensina na computação é que todo problema passa por três estágios: Entrada; Processamento e Saída.

Para o estudante de engenharia civil a Entrada pode ser as dimensões de uma viga, o Processamento uma norma específica da ABNT, e a Saída a quantidade de cimento e ferro necessária para que a viga agüente o esforço a ela destinado.

Para um estudante de engenharia elétrica a Entrada pode ser o consumo de energia por unidade familiar, o Processamento o dimensionamento de uma linha de transmissão, e a Saída um projeto de toda uma infra-estrutura de distribuição.

Analogamente o estudante de engenharia mecânica pode estar preocupado em projetar um braço de um guindaste, um estudante de engenharia química uma caldeira e o aluno de engenharia Eletrônica esteja preocupado com um chip específico.

Se o assunto é da competência de um engenheiro e o tema em questão possui alguma norma descrita por um organismo sério de normas e padrões técnicos, então provavelmente o assunto poderá ser descrito e automatizado em forma de um programa.

Esse é o papel da Computação do curso de engenharia: ensinar "boas práticas" de raciocínio lógico-matemático que podem ser até programadas em alguma calculadora ou computador.

Histórico da disciplina na EEUPM

É prudente ressaltar como a disciplina evoluiu nos últimos 20 anos, destacando-se alguns aspectos curiosos que estavam ocorrendo na época de cada mudança.

Em 1984 os alunos da Escola de Engenharia da Universidade Mackenzie aprendiam FORTRAN, faziam fluxogramas e digitavam seus códigos em cartão perfurado. Nesta época começavam a chegar no Brasil os primeiros computadores pessoais de 8 bits com seus primeiros softwares aplicativos.

Os alunos do ano de 1985 já tiveram suas aulas de computação com a linguagem BASIC. Ainda usavam fluxograma como ferramenta de trabalho e não puderam testar seus programas na Universidade, pois não havia computadores suficientes para atender a demanda.

O problema de laboratório de informática deficitário persistiria por alguns anos, o que fez os professores adotarem em 1986, para a disciplina de computação, a linguagem Algorítmica, uma interessante linguagem estruturada de programação que ilustrava didaticamente bem a funcionalidade do computador. Tal estratégia provou ser possível ensinar computação sem que os alunos tivessem acesso ao computador.

Em 1987 os professores de computação optaram por introduzir a linguagem PASCAL na disciplina em conjunto com os exercícios de linguagem Algorítmica.

No início dos anos 90 a IBM lança seu padrão de computador pessoal, o PC-XT, que inicia uma série de mudanças nos programas disponíveis no mercado.

Em 1991 na disciplina de Computação apresenta-se aos alunos a linguagem PASCAL e uma coletânea de aplicativos, entre eles: editores de texto, planilhas e CAD. No fim do ano há uma certa pressão por parte dos alunos e até de alguns professores, que desejavam reformular as ementas do curso de engenharia, em se abandonar o ensino de qualquer linguagem de programação para apenas se ensinar bem como usar aplicativos que começavam a inundar o mercado. Da polêmica conclui-se que era mais importante trabalhar a lógica e deixar que o aluno fosse autodidata nos softwares relevantes.

Ensina-se PASCAL até 1994, quando a empresa Borland lança a 1ª versão do Delphi, que funcionava sobre o Windows 3.1. Neste este ano, em uma atitude ousada, os professores por iniciativa própria adotaram esta nova ferramenta de programação. O curso foi adequado rapidamente. Com a adoção de um ambiente visual de programação para se ensinar descobriu-se uma forma diferente de envolver os alunos na disciplina. A ferramenta visual cativava o aluno, que já estava cansado da tela escura com caracteres como interface.

Por mais de 6 anos os professores desenvolvem a disciplina de computação com a ferramenta Delphi, o que faz surgir uma série de trabalhos bem elaborados, além de se conquistar respeito e admiração por parte dos alunos por se estar obtendo resultados tão rápidos.

No fim de 2000 o Departamento de Eletrônica solicita que as disciplinas de Computação façam uso da linguagem C ou C++. Isso no início foi um impasse, pois toda a metodologia estava desenvolvida sobre a plataforma visual Delphi.

Coincidentemente em 2001 começa a ficar popular no mercado Brasileiro uma versão do C++Builder, um produto similar ao Delphi que faz uso da linguagem C++ para o desenvolvimento de programação.

Desde o ano de 2002 adota-se nas disciplinas de Computação, da Escola de Engenharia Mackenzie, o C++Builder como ferramenta de ensino.

Curiosamente, nessas duas últimas décadas de ensino da disciplina, o que se manteve constante foram os conteúdos programáticos, os exercícios matemáticos desenvolvidos ao longo do período. Do FORTRAN ao C++Builder, do cartão

perfurado ao computador conectado à Internet, os exercícios matemáticos desenvolvidos foram a base dos trabalhos para se moldar nos alunos a lógica do engenheiro.

A resolução dos problemas mostrou-se mais importante que a própria linguagem de programação em voga, ou mesmo o hardware mais poderoso. A linguagem de programação e o modelo de computador apresentaram-se apenas como "ferramentas" do processo de aprendizagem. O "pensar" nos problemas propostos mostrou-se ser a verdadeira contribuição da disciplina.

É na disciplina de Computação que o aluno começa a pensar como engenheiro!

Literatura técnica da disciplina

Até 1984 os livros encontrados nas livrarias, que tratavam de computação, se preocupavam mais em ensinar como elaborar programas do que como apertar botões e o que estava disponível nas telas dos aplicativos.

Com a disseminação massiva dos computadores pessoais esses livros, verdadeiras pérolas de saber computacional, começaram a rarear nos lançamentos literários.

Com essa preocupação em mente, os autores do presente artigo, têm procurado manter um pouco deste conhecimento circulando (fizemos isso mesmo quando éramos alunos!).

Entre 1986 e 1997 cada um dos autores, em separado ou em conjunto com outros colegas inclusive, disponibilizou para seus alunos e amigos alguma forma de apostila que ressaltasse os exemplos e exercícios de programação para resolução dos complexos problemas matemáticos abordados no curso de Engenharia.

Entre 1997 e 1999 abraçamos a empreitada de compilar nossas anotações de aula em nosso primeiro livro em conjunto, "Delphi para Universitários" [1], que já está com a sua 3ª edição no prelo.

Em 2002, conforme as necessidades da nossa Escola de Engenharia, revisamos e melhoramos o texto didático para lançar, também em conjunto, o livro "C++ Builder para Universitários" [2], que já se encontra em sua 2ª edição.

METODOLOGIA

É importante mostrar ao aluno como o computador também pode ser uma ferramenta muito útil para o desenvolvimento dos seus cálculos complexos durante o próprio curso de engenharia.

Cativando o aluno

É muito mais agradável, para o aluno, ver o resultado de um programa em um formulário com botões e campos com aparência tridimensional do que tentar ler e entender mensagens de texto de uma tela alfanumérica.

O método adotado procura contemplar uma combinação entre a facilidade da ferramenta visual na criação dos campos no formulário com a rápida edição direcionada do programa.

As ferramentas de programação com recursos visuais podem ser um diferencial no processo de ensino na lógica de programação. Enquanto o aluno fica com a curiosidade aguçada para saber qual será a próxima novidade a ser apresentada.

As representações da lógica

Em épocas distintas a mesma lógica do engenheiro foi representada de várias maneiras diferentes. Há uma geração de engenheiros que aprendeu a representação da lógica por meio de fluxogramas, semelhantes ao padronizado na norma ISO 5807. Um outro grupo de profissionais teve seu raciocínio lógico-matemático moldado por alguma forma de algoritmo redigido por meio de uma linguagem simples, um português técnico. E, mais recentemente, um outro grupo de engenheiros foi acostumado a representar suas idéias através de diagramas N-S, que foram desenvolvidos por Ike Nassi e Ben Shneiderman (daí a origem do nome N = Nassi e S = Shneiderman).

Embora o número de aulas atribuído às disciplinas de Computação nos cursos de Engenharia normalmente seja reduzido, sempre é possível se fazer alguma referência às três formas de representação da lógica, seja em uma aula específica para tratar o assunto, ou seja, no meio das explicações dos mais variados comandos abordados.

A diversidade de representações lógicas também apresenta um efeito colateral interessante no aprendizado. Dependendo do aluno, ou dos recursos disponíveis nas aulas, certas representações podem surtir melhor efeito do que outras, ou seja, pode-se ensinar a lógica com a representação que apresentar melhor receptividade ao público alvo.

A matemática explorada pela disciplina

A lógica do engenheiro é moldada em um universo de cálculos matemáticos tratados de forma numérica. Exploram-se muito o uso de funções matemáticas, integrações numéricas, matrizes e vetores, séries matemáticas, entre tantos outros assuntos. O estudante tem que entender o problema matemático e conhecer os passos para encontrar sua solução, ou seja, sua lógica (ou algoritmo de resolução).

Parte dos temas matemáticos, abordados nas disciplinas de Computação, está explicado no livro "Calculo Numérico para Universitários" [3], material texto da disciplina de Cálculo Numérico.

Em outras palavras, o Cálculo Numérico pode ser entendido, pelo aluno, como um ferramental matemático para entender e tratar um problema de outra disciplina (p.ex: Física). Uma vez dominados os métodos de cálculo, nessa outra disciplina, o ferramental é introduzido na seqüência lógica da solução de um problema maior, um problema de

engenharia (p.ex: cálculo de esforços por elementos finitos). Cabe ao aluno, no momento de elaborar seu programa, entender tanto o ferramental quanto o problema de engenharia para codificar a seqüência correta de comandos no código do software.

Uma seqüência ilustrativa de atividades

Para apresentar melhor a metodologia, de como se partir de conceitos simples para poder se explicar programas de cálculos complexos, a seguir apresentam-se alguns temas relacionados que são desenvolvidos seqüencialmente durante o andamento das aulas, em um ou dois semestres:

Etapa I

Em ocasião apropriada, o conceito de funções pré-programadas é apresentado aos alunos. Explica-se a necessidade de se declarar uma unidade, "biblioteca (1)", que contenha a função, "função (2)" e "função (3)", previamente programada pelo fabricante da linguagem.

```
# include <math.h>        (1)
...
y=sin(x);                 (2)
z=cos(x);                 (3)
```

Etapa II

Busca-se ensinar comandos condicionais, "comando (4a)" e "comando (4b)", e um ou mais comandos de repetição, "comando (5a)" e "comando (5b)", usando tanto a linguagem adotada quando os instrumentos de representação da lógica, "algoritmo (6)" "algoritmo (7a)" e "algoritmo (7b)".

```
if (...) { ... } else { ... }      (4a)
switch (...){ case: ... ; ... }    (4b)
for (i=1; i<=n; i++) {...}         (5a)
while (qtd<Maximo) {...}           (5b)
Se (...) Então ... Senão ...       (6)
Para I←1 até N faça ...            (7a)
Enquanto (Qtd<Maximo) ...          (7b)
```

Etapa III

Tanto na disciplina de Computação como na disciplina de Cálculo Numérico é apresentado ao aluno o conceito que as funções podem ser transformadas em polinômios. A série do seno, "série (9)", e do cosseno, "série (10)", podem ser deduzidas pela fórmula de Maclaurin, "fórmula (8)".

$$f(x) = f(0) + f'(0)\frac{x}{1!} + f''(0)\frac{x^2}{2!} + f'''(0)\frac{x^3}{3!} + \ldots \qquad (8)$$

$$f(x) = Seno(x) = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \ldots \qquad (9)$$

$$f(x) = Cosseno(x) = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \frac{x^6}{6!} + \ldots \qquad (10)$$

Etapa IV

Com os comandos de repetição pode-se mostrar aos alunos que as séries matemáticas apresentam os mesmos resultados que as funções de biblioteca da linguagem, "figura (I)".

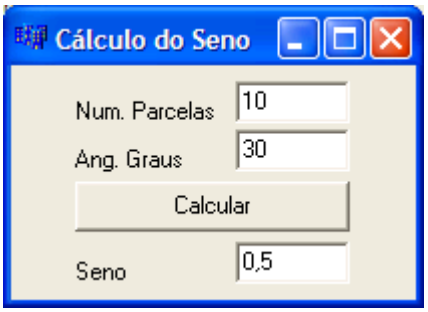

FIGURA I
PROGRAMA QUE CALCULA A FUNÇÃO SENO.

A seguir segue a codificação da "equação (9)" na "figura (II)", com comentários para melhor compreensão dos alunos.

```
void __fastcall TForm1::Button1Click(TObject *Sender)
{
  int i, j, n, t;
  double x, f, s;
  n=StrToInt(Edit1->Text);        //Número de parcelas
  x=StrToFloat(Edit2->Text);      //Ângulo em graus
  x=x*M_PI/180.0;                 //Converte x em radianos
  s=0.0;                          //Inicia a série
  t=1;                            //Trocador de sinal
  for (i=1;i<=n;i++)
    {
     f=1.0;
     for (j=1;j<=2*i-1;j++)
       f=f*j;                     //Calcula o fatorial
     s=s+t*pow(x,2*i-1)/f;        //Acumulo da parcela
     t=-t;                        //Troca o sinal
    }
  Edit3->Text=FloatToStr(s);      //Seno do ângulo x
}
```

FIGURA II
PROGRAMA QUE CALCULA A FUNÇÃO SENO.

Etapa V

A possibilidade de converter programas em funções é algo muito interessante.

Transmite-se ao aluno que seus próprios programas poderão ser reusados como rotinas, funções, em outros programas.

Nesta fase também se explora pequenas melhorias no código para que o mesmo execute os mesmos cálculos com menos trabalho do computador. A seguir segue a codificação da "equação (10)" em forma de função, na "figura (III)".

```
double Cosseno(double x)   //Entra angulo e sai Cosseno
{
  double p, s, erro, q;
  int k=1;
  x=x*M_PI/180.0;          //Converte x em radianos
  p=1.0;
  s=1.0;
  erro=0.0001;
  q=x*x;
  while(fabs(p)>erro)      //Enquanto o erro for grande
    {
     p=-p*q/k/k+q;         //Calcula a série
     s=s+p;
     k=k+2;
    }
  return s;                //Devolve a resposta
}
```

FIGURA III
TEXTO PROGRAMA DA FUNÇÃO COSSENO.

Etapa VI

Aproveitando que os alunos também passam por aulas de Cálculo Diferencial e Integral, ensina-se na Computação como se podem usar recursos gráficos gerados a partir de programas relativamente simples. Nesta etapa se explora o desenvolvimento de programa para o cálculo de integrações numéricas, conforme a "figura (IV)".

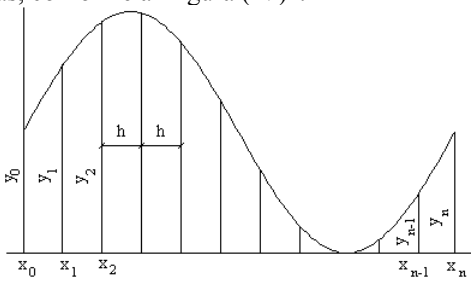

FIGURA IV
EXPLICAÇÃO DA INTEGRAÇÃO NUMÉRICA.

$x_0 = a$ (11)
$x_n = b$ (12)
$h = (b-a)/n$ (13)
$x_i = x_0 + i*h$ (14)
$y_i = Sen(x_i) + 1$ (15)
$Área = h\ (y_0/2 + y_n/2 + y_1 + y_2 + \ldots + y_{n-1})$ (16)

Onde a "equação (11)" e "equação (12)" delimitam os limites da integral, a "equação (13)" define o passo de cálculo da integração, a "equação (14)" calcula como se fosse uma P.A. (Progressão Aritmética) os valores do eixo X relevantes, a "equação (15)" é a própria função a ser integrada, e a "equação (16)" é a fórmula da integração pelo método dos trapézios.

Etapa VII

Da disciplina de Álgebra Linear aproveita-se a teoria que ensina a redução de coordenadas espaciais (P(x',y',z')) em coordenada no plano (P(x,y)), pelo método R3R2. Resumidamente R3R2 significa aplicar nas equações abaixo, "equação (17)" e "equação (18)", uma coordenada espacial em um sistema cartesiano sendo observado segundo uma configuração angular; ângulos: Wx, Wy e Wz. Nas equações: cx=CosWx, cy=CosWy, Cz=CosWz, sx=SenWx, sy=SenWy, sz=SenWz, a=sx.sy.

$x = x_t + (cy.cz).x' - (cy.sz).y' + sy.z'$ (17)
$y = y_t + (a.cz+cx.sz).x' + (-a.sz+cx.cz).y' - (sx.cy).z'$ (18)

Uma vez explicado o ferramental algébrico aproveita-se para serem apresentados os conceitos de Classes e Objetos, e então criar uma classe TPonto3D que poderá ser empregada em exercícios mais complexos.

```
class TPonto3D
{
private:
 double x3d,y3d,z3d,x2d,y2d;
 double wx,wy,wz,xt,yt,zoom;
 int    col,lin;
 void R3R2();
 void Ajuste();
 void Converte();
public:
 void SetP3d(double x, double y, double z,
  double ax=110,double ay=180,double az=60,
  double dx=0,double dy=0,double zm=1);
 int GetCol();
 int GetLin();
};
```

FIGURA V
CLASSE TPONTO3D.

Etapa VIII

Elaborada a classe TPonto3D pode-se, então, propor aos alunos o desenvolvimento de programas mais complexos, "figura (VI)", que servem para ilustrar problemas de outras disciplinas do curso de engenharia.

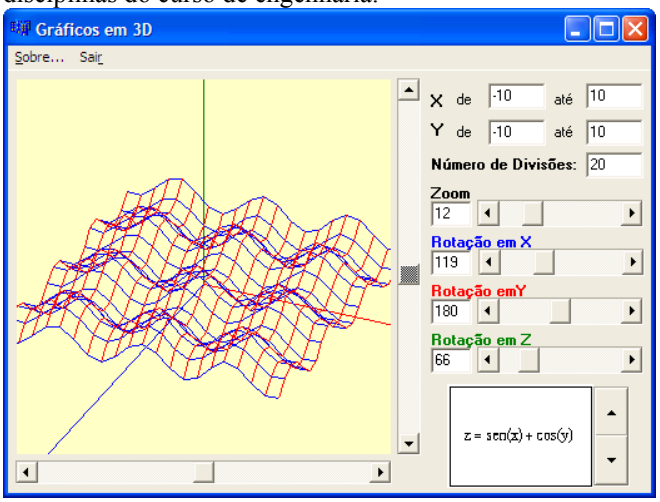

FIGURA VI
PROGRAMA QUE ILUSTRA A FUNÇÃO Z=SEN(X)+COS(Y).

No exemplo da "figura (VI)" o aplicativo foi criado com uso da classe TPonto3D e também foram empregadas funções que foram criadas pelos alunos, analogamente ao que foi visto na "série (9)" e na "série (10)".

CONSIDERAÇÕES FINAIS

A nossa maior preocupação é que o atual conhecimento abordado na disciplina de Computação, que permite calcular as mais variadas funções, sejam elas complexas ou não, não se perca com a sedução de usar os "aplicativos enlatados" que cada vez mais aparentemente resolvem tudo.

Quando o conhecimento se perde (como, por exemplo, o uso da régua de cálculo) o seu resgate pode se tornar custoso, passar a ser domínio de umas poucas empresas que o confinam em chips de computadores (verdadeiras "caixas pretas") que serão vendidos a elevados preços.

Recomendamos que cada colega professor busque registrar as pequenas pérolas do saber de suas disciplinas, antes que seja tarde.

REFERÊNCIAS BIBLIOGRÁFICAS

[1] BARROS, Edson de Almeida Rego; PAMBOUKINAN, Sergio Vicente D.; ZAMBONI, Lincoln César. "Delphi para Universitários" - 2ª Edição. São Paulo: Páginas & Letras Editora e Gráfica, 2000.

[2] BARROS, Edson de Almeida Rego; PAMBOUKINAN, Sergio Vicente D.; ZAMBONI, Lincoln César. "C++ Builder para Universitários" - 2ª Edição. São Paulo: Páginas & Letras Editora e Gráfica, 2003.

[3] ZAMBONI, Lincoln César; JÚNIOR, Orlando Monezzi. "Cálculo Numérico para Universitários" - 2ª Edição. São Paulo: Páginas & Letras Editora e Gráfica, 2002.

[4] BARROS, Edson de Almeida Rego; PAMBOUKINAN, Sergio Vicente D.; ZAMBONI, Lincoln César. Artigo apresentado no COBENGE2004 (CONGRESSO BRASILEIRO DE ENSINO DE ENGENHARIA), Brasília – DF, Unb 14 a 17 set 2004.