

ALGbr: Uma Nova Ferramenta Para Apoio ao Ensino/Aprendizagem de Lógica Computacional por Meio da Construção e Testes de Algoritmos

Carlos J. Santos¹, Matheus E. Franco¹, Hagar C. C. Corsini¹

¹Área de Computação - IFSULDEMINAS- Campus Machado
CEP - 37.750-000 – MG – Brasil

cakynho@gmail.com, {matheus.franco,hagar}@mch.ifsuldeminas.edu.br

Abstract. *The representation of algorithms by means of the software enables agility in writing and review, facilitating debugging, verification mechanisms providing ha possible flaws in the writing, enabling the repurposing and reusing code, making changes and updates, favoring the understanding and deepening the student's perception, through the experimentation and matching of theory combined with practice in tests with algorithms. The new proposed tool shows itself as an alternative to the other, by the fact that the use of the construction of C-based algorithms in addition to the traditional approach based on PASCAL.*

Resumo. *A representação dos algoritmos por meio do software viabiliza a agilidade na escrita e na revisão, facilitando a depuração, fornecendo mecanismos de verificação ha possível falhas na escrita, possibilitando o reaproveitamento e reutilização de código, facilitando as alterações e atualizações, favorecendo o entendimento e aprofundando a percepção do aluno, através da experimentação e congruência da teoria aliada à prática nos testes com algoritmos. A nova ferramenta proposta mostra-se como uma alternativa as demais existentes, pelo fato de possibilitar a utilização da construção de algoritmos baseados em linguagem C além da abordagem tradicional baseada em PASCAL.*

1. Introdução

A disciplina de algoritmos é considerada por professores e alunos de difícil compreensão. Na construção de algoritmos, para solução de problemas simples é necessário entendimento e raciocínio lógico assaz. Habitualmente o algoritmo é feito em uma folha em branco, onde o aluno muitas vezes não consegue expressar seu raciocínio baseado nos conceitos adquiridos ou a correta grafia mesmo quando escrita em linguagem natural.

A utilização do pseudocódigo ajuda a quebrar a dificuldade na expressão das ideias, tornando a escrita mais próxima das utilizadas nas linguagens de programação, entretanto o teste de mesa, nome dado a “compilação” das informações expressadas no papel apresenta limitações.

Especialistas têm experimentado várias abordagens no ensino de algoritmos, conceito de difícil apreensão, pois que não basta apresentar um algoritmo pronto para que o aluno seja capaz de criar outro, em situações parecidas. (BARBOSA, 2001).

Perante a dificuldade do processo de ensino-aprendizagem, mesmo diante de problemas simples, evidencia-se a necessidade da utilização de um software de apoio didático-pedagógico. Neste contexto, a utilização do software de apoio didático-pedagógico, desenvolvido em um ambiente mais atraente que uma folha de papel, com recursos de visualização concreta do que se está sendo criado contribui para o ensino/aprendizagem da lógica computacional.

[...] não há apreensão conceitual sem a coordenação de vários sistemas de representação semiótica. (DUVAL, 1995).

Existem no mercado ferramentas similares, mas cada uma com suas peculiaridades de sintaxe e estrutura de desenvolvimento. A exemplo pode-se citar o Visualg (2012), software que edita, interpreta e executa algoritmos com uma linguagem próxima do português estruturado e o Portugol, software muito utilizado para o ensino da lógica de programação e algoritmos, presente em muitos materiais didáticos de programação em português sobre o assunto, mas sem um padrão sintático ou léxico bem definido.

O que diferencia o projeto do ALGbr dos outros projetos relacionados é a preocupação de aproximar ao máximo sua sintaxe da utilizada nos algoritmos escritos de forma manual. Para isso conta com a possibilidade de escrita de algoritmos em um padrão semelhante à linguagem de programação PASCAL, um dos padrões mais difundidos no ensino de programação.

Porém, somente a didática e facilidade de aprendizado proporcionado por esta sintaxe baseada em Pascal não garante que o aprendiz possa obter resultados significativos quando da necessidade da real escrita de um software, principalmente se levar em consideração os padrões atuais de desenvolvimento. Deve-se observar que a linguagem PASCAL e suas derivadas encontram-se respectivamente na 15ª e 18ª posições no ranking da TIOBE Programming Community Index (2013) (lista ordenada de linguagens de programação, classificada pela frequência de pesquisa na web usando o nome da linguagem como palavra-chave). De acordo com o site a frequência de buscas pode refletir o número de engenheiros hábeis, cursos e vagas de emprego no mundo todo.

Perante esta visão, a busca por uma sintaxe que ao mesmo tempo trabalhe o aprendizado da lógica, mas que também capacite o aluno a mais utilizada linguagem de programação e suas derivações, que há muito tempo são padrões de mercado (pelo menos as cinco primeiras posições do ranking se baseiam em C), haverá então a possibilidade de escrita dos algoritmos no padrão C.

Para Manzano (2011) a construção de um algoritmo permuta entre duas fases: concepção e representação. A representação dos algoritmos por meio do software viabiliza a agilidade na escrita e na revisão, facilitando a depuração, fornecendo mecanismos de verificação ha possível falhas na escrita, possibilitando o reaproveitamento e reutilização de código, facilitando as alterações e atualizações, favorecendo o entendimento e aprofundando a percepção do aluno, através da experimentação e congruência da teoria aliada à prática nos testes com algoritmos.

2. O Ensino de Algoritmos

Segundo Berg & Figueiró (1998) um algoritmo pode ser descrito basicamente de duas formas: uma gráfica a partir da utilização de diagramas de blocos, e outra textual a partir

de uma linguagem de projeto de programação ou mesmo de uma linguagem de programação de computadores formal. Num sentido mais amplo, algoritmo é um processo sistemático para a resolução de um problema (SZWARCFITER & MARKENSON, 1994), ou de uma sequência ordenada de passos a ser observada para a realização de uma tarefa (SALIBA, 1992) e (BERG & FIGUEIRÓ, 1998). Para Berlinsk (2002), algoritmo é um método finito, escrito em um vocabulário simbólico fixo regido por instruções precisas, que se movem em passos discretos, 1, 2, 3,..., cuja execução não requer insight, esperteza, intuição, inteligência ou clareza e lucidez, e que mais cedo ou mais tarde chega a um fim. Para Cormen *et al.* (2002), um algoritmo é qualquer procedimento computacional bem definido que toma algum valor ou conjunto de valores como entradas e produz algum valor ou conjunto de valores de saída.

Raabe (2005) considera fundamental o aprendizado de algoritmos, onde sua prática está diretamente ligado ao raciocínio lógico. Por ventura, este ganho é utilizado no decorrer de todos os cursos da área computacional.

O algoritmo não se forma simplesmente com uma sintaxe descritiva desordenada, o raciocínio lógico é que determinará a sequência correta para que os passos a serem seguidos fluam até o retorno esperado. Mas o raciocínio lógico vai além, pois inúmeros fatores são necessários para completá-lo, tais como conhecimento, versatilidade, experiência, criatividade, responsabilidade, ponderação, calma, autodisciplina, entre outros. Sua utilização demanda domínio do pensar, coerência.

[...] lógica é a ciência que estuda as leis e os critérios de validade que regem o pensamento e a demonstração, ou seja, ciência dos princípios formais do raciocínio, tendo sido possivelmente descoberta por Aristóteles, que viveu entre os anos de 384 e 322 a.C. (ABE & SCALZITTI, 2001).

Para Fosnot (1998) o processo de ensino-aprendizagem construtivista aborda os princípios de que aprender é desenvolvimento e que a força motora da aprendizagem é a reflexão sobre as representações construídas.

Cormen *et al.* (2002) afirma que uma sólida base de conhecimento e técnica de algoritmos é uma característica que separa os programadores verdadeiramente qualificados dos novatos. Com a moderna tecnologia computacional, você pode executar algumas tarefas sem saber muito sobre algoritmos; porém, com uma boa base em algoritmos, é possível fazer muito mais.

3. Materiais e Métodos

Neste projeto, a ferramenta ALGbr, utilizada na transição da linguagem natural em pseudocódigo se encaixa na categoria de tradutor – o programador escreve o algoritmo em pseudocódigo e ela reescreve na linguagem em que o compilador possa gerar um programa propriamente dito. O processo de compilação é realizado nos algoritmos escritos no padrão PASCAL pelo compilador FPC (Free Pascal Compiler) e para os padronizados em C o compilador GCC (GNU Compiler Collection), ambos open-source sob a licença GPL, que acompanharão o software.

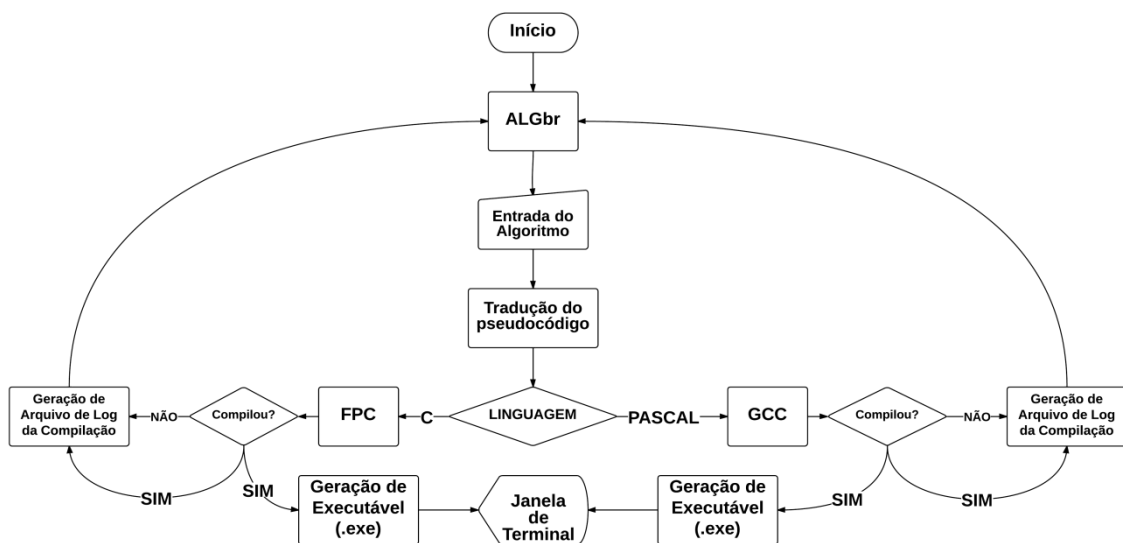


Figura 1. Fluxograma de funcionamento da ferramenta

As sintaxes utilizadas para interpretação dos algoritmos baseados em PASCAL e C são apresentadas nas figuras 2 e 3.

```

01 algoritmo novo
02
03 declare
04 a : inteiro
05 b : inteiro
06 i : inteiro
07 t : inteiro
08
09 inicio
10 a:=0
11 b:=0
12 i:=0
13 t:=0
14
15 escreva("digite o primeiro valor: ")
16 leia(a)
17 escreva("digite o segundo valor: ")
18 leia(b)
19 t:=a+b
20
21 se (t>10) entao
22   para t:=1 ate t faça
23     escreva(i," ")
24   fimpara
25 fimse
26
27 fimalgoritmo
  
```

Figura 2. ALGbr: Padrão PASCAL

```

01 algoritmo novo {
02
03 inteiro : a=0;
04 inteiro : b=0;
05 inteiro : t=0;
06 inteiro : i=0;
07
08 escreva("digite o primeiro valor: ");
09 leia(a);
10 escreva("digite o segundo valor: ");
11 leia(b);
12
13 t:=a+b;
14
15 se (t>10) {
16   para (i=1; i<=t; i++) {
17     escreva(i, " ");
18   }
19 }
20 }
  
```

Figura 3. ALGbr: Padrão C

Há uma interoperabilidade entre interpretador e compilador, abstraindo do usuário o processo de conversão e compilação. O que fica claro é a visão do algoritmo sendo executado como um programa em modo terminal, fornecendo assim um resultado visível de sua implementação.

Todos os erros de sintaxe e semântica são tratados pelo compilador, mas exibidos no interpretador, que por sua vez informa a linha e coluna do eventual erro.

No ato da digitação as palavras reservadas são destacadas, melhorando a distinção da codificação. Há a possibilidade de escrita de comentários ao longo do algoritmo possibilitando criar anotações ou lembretes que possam favorecer uma releitura posterior do pseudocódigo. A produção do software segue os preceitos da arquitetura de aplicações para sistemas de processamento de linguagens. Esta arquitetura aceita uma linguagem natural ou artificial como entrada e gera alguma outra representação dessa linguagem como saída.

O desenvolvimento do software foi realizado na linguagem C#, utilizando a IDE Visual Studio 2010 adquirida por meio da aliança acadêmica *DreamSpark*.

4. Resultado e Discussão

Através de testes e avaliações por parte de dois professores com experiência em ensino de algoritmos, foram relatados resultados foram satisfatórios, pois os mesmos declararam que a interface segue uma estrutura intuitiva e de fácil operação, o que veio a cativar seu uso pelos alunos. Os professores avaliadores estão levando a ferramenta a campo, utilizando-a em sala de aula. Sua eficácia virá a ser mensurada no decorrer do ano letivo, aplicando-a como recurso didático e avaliativo.



Figura 4. ALGbr: Ambiente de escrita

Para validação da ferramenta, também será realizado um teste com os alunos em formato de pesquisa experimental, subdividindo uma turma de alunos que ainda não tem convivência com algoritmos e linguagens de programação em três grupos. No primeiro estágio irá aplicar-lhes aula expositiva sobre a dinâmica dos algoritmos, seguindo com apresentação da ferramenta ALGbr, explicando suas funcionalidades.

A divisão dos grupos dará da seguinte forma: Grupo1 – Aplicação de pseudocódigo baseado em PASCAL no ALGbr; Grupo 2 – Aplicação de pseudocódigo baseado em C no ALGbr (Grupo de Controle); Grupo 3 – Aplicação de pseudocódigo no Visualg (em sua linguagem padrão).

Para clarificar o estudo, será apresentada aula expositiva da sintaxe da linguagem C, realizando-se uma comparação entre o pseudocódigo baseado em C, PASCAL e na linguagem nativa do Visualg com o mesmo código. Haverá uma explanação entre as diferenças nos códigos e então aplicar-se-á aos alunos um questionário que medirá as dificuldades ou não enfrentadas pelos alunos, estes dados serão transformados em estatísticas.

Baseado na hipótese de que o uso de algoritmos baseados na linguagem C serão mais bem recebidos pelos alunos, quando comparadas as outras linguagens, esperamos que a aceitação desse pseudocódigo tenha uma maior aceitação. O cronograma de testes será realizado conforme apresentado na tabela 1.

Tabela 1 - Cronograma de Testes

CRONOGRAMA DE TESTES COM ALUNOS DA FERRAMENTA ALGBR			
	GRUPO 1	GRUPO 2 (CONTROLE)	GRUPO 3
FASE 1	Aula Expositiva Algoritmos		
FASE 2	Apresentação do ALGbr	Apresentação do ALGbr	Apresentação do Visualg
FASE 3	Pseudocódigo baseado em PASCAL	Pseudocódigo baseado em C	Pseudocódigo baseado em linguagem nativa
FASE 4	Aula Expositiva de Linguagens baseadas em C		
FASE 5	Comparação dos pseudocódigos elaborados com o código na linguagem C.		
FASE 6	Aplicação de Questionário		
FASE 7	Tabulação e Análise de Dados		
Fonte: Elaborado pelo autor			

Encontra-se em desenvolvimento rotinas que criam critérios para customização do software, com a finalidade de aumento gradativo da complexidade dos pseudocódigos. De acordo com as considerações apontadas pelos professores avaliadores novos recursos poderão ser absorvidos pelo software.

5. Considerações Finais

Levando-se em consideração as limitações das ferramentas existentes para o ensino de algoritmos, no que tange a linguagem utilizada, o sistema proposto supre necessidades existentes como a possibilidade de parametrizações de uso de delimitadores de final de linha de comando, *intelligent sense* (preenchimento automático durante a digitação), destaque de palavras reservadas, ajuste da área de digitação (fonte, cores, tamanho), inclusão de comentários no algoritmo, zoom facilitando a apresentação em Datashow, painel de ocorrências (demonstra o retorno do compilador informando o(s) erro(s) e a(s) linha(s) com incoerência(s) caso exista(m)), tempo de compilação, numeração das linhas, sistema de ajuda (perguntas frequentes, sintaxe dos pseudocódigos, palavras reservadas, etc.) e execução passo a passo.

A ferramenta proposta mostra-se como uma alternativa as demais existentes, pelo fato de possibilitar a utilização da construção de algoritmos baseados em linguagem C além da abordagem baseada em PASCAL.

Ao atingir uma versão estável, o código fonte do ALGbr será disponibilizado sob licença GPL para que haja continuidade em seu desenvolvimento e aprimoramento pela ajuda da comunidade focado no assunto (professores, estudantes, etc.).

O próximo passo de desenvolvimento será a concepção de um compilador próprio que possa englobar as duas ou mais linguagens que o software possa dar suporte.

6. Referências Bibliográficas

- ABE, J. M.; SCALZITTI, A., SILVA FILHO, J. I. Introdução à Lógica para Ciência da Computação. São Paulo: Arte e Ciência, 2001.
- BARBOSA, L. M. Ensino de Algoritmos em Curso de Computação. São Paulo: Editora EDUC, 2001.
- BERG, A. C.; FIGUEIRÓ, J. P. Lógica de Programação. 2. Ed. Rio Grande do Sul: ULBRA, 2002.
- BERLINSK, D. O Advento do Algoritmo: A Idéia que Governa o Mundo. São Paulo: Globo, 2002.
- CORMEN, T. H.; LEISERSON C. E.; RIVEST R. L. Stein C. Algoritmos: Teoria e Prática. Rio de Janeiro: Elsevier, 2002.
- DUVAL, R. Sémiosis et pensée humaine – registres semiótiques et apprentissages intellectuels. Berne, Peter Lang, 1995.
- FPC - Free Pascal Compiler. Disponível em: <<http://www.freepascal.org/>>. Acesso em Abril/2013.
- FOSNOT, C. T. Construtivismo – teoria, perspectivas e prática pedagógica. Tradução de Sandra Costa. Porto Alegre, ArtMed, 1998.
- GCC - GNU Compiler Collection. Disponível em: <<http://gcc.gnu.org/>>. Acesso em Abril/2013.
- MANZANO J. A. N. G.; OLIVEIRA J. F.; Algoritmos – Lógica para Desenvolvedores de Programação de Computadores, 25ª Edição, São Paulo, Editora Érica, 2011.
- PORTUGOL. Disponível em: <<http://www.dei.estt.ipt.pt/portugol/>>. Acesso em Abril/2013.
- RAABE, André Luís Alice, Uma proposta de arquitetura de Sistema Tutor Inteligente baseada Teoria das Experiências de Aprendizagens Medianas – Tese de Doutorado, 2005.
- SALIBA, W. L. C. Técnicas de Programação: Uma Abordagem Estruturada. São Paulo: Makron Books 1993.
- SZWARCFITER, J. L.; MARKENZON, L. Estruturas de Dados e Seus Algoritmos. 2. Ed. Rio de Janeiro: Livros Técnicos e Científicos, 1994.
- TIOBE Programming Community Index for April 2013. Disponível em: <<http://www.tiobe.com/index.php/content/paperinfo/tpci/index.html>>. Acesso em Abril/2013.
- VISUALG. Disponível em: <<http://www.apoioinformatica.inf.br/visualg/>>. Acesso em Dezembro/2012.