

SUMÁRIO

1	INTRODUÇÃO	2
1.1	OBJETIVOS DO TRABALHO	3
1.1.1	Objetivo Geral	3
1.1.2	Objetivos Específicos	3
1.2	JUSTIFICATIVA DO TRABALHO	3
1.3	ESTRUTURA DO TRABALHO	4
2	FUNDAMENTAÇÃO TEÓRICA	5
2.1	ENSINO APRENDIZAGEM DE ALGORITMOS EM PORTUGOL	5
2.1.1	O Ensino de Algoritmos	6
2.2	O ENSINO DA LINGUAGEM DE PROGRAMAÇÃO PORTUGOL	8
2.2.1	A Ferramenta Portugol/Plus	10
2.3	PROCESSAMENTO DE LINGUAGEM NATURAL	13
	REFERÊNCIAS	14

1 INTRODUÇÃO

A base da programação é ensinar máquina a realizar determinada tarefa. Para isso pode-se dizer que independente do paradigma utilizado no processo de desenvolvimento ele envolve algoritmos.

Algoritmos podem não necessariamente serem computacionais, já que são um conjunto de passos finitos. Analogias simplistas são os ingredientes e modo de preparo de uma receita ou interiores para uma tarefa do cotidiano como a troca de um pneu furado em um automóvel.

Porém se tratando de computação deve ter maior exatidão seguindo uma linguagem pois diferente do ser humano uma máquina não é tão interpretativa. Desconsideradas aqui inteligências artificiais que são desenvolvidas por algoritmos para terem comportamento de compreensão humana. Isto pode ser analogamente algo como não entender idioma estrangeiro e as instruções estarem nesta linguagem. Porém em informática isso tem que ser mais especificado com palavras ou caracteres predeterminados de início e fim, de blocos, repetições, etc.

Sendo o berço mais popular da informática os Estados Unidos da América e seus criadores em muitos casos dessa origem ou de países com mesmo idioma as primeiras linguagens de computação e a maioria delas são em inglês.

Isso muda quando falamos do Portugal que possui variações, mas resumidamente sua origem vem como uma tradução de uma linguagem de programação simples e comum com seus termos em nosso idioma nativo no Brasil, pela nossa colonização portuguesa.

Tendo como foco o aprendizado de algoritmos para uma boa fundamentação de futuros programados ou profissionais da área, e muitas vezes a facilidade com o inglês não é algo comum nosso país, a linguagem em Portugal de torna um artifício vantajoso.

O que se propõe aqui em ter ferramenta de aprendizado composta de editor de texto para a escrita dos algoritmos e visualizador do resultado dos mesmo. E que não seja necessária instalação, seja facilmente acessível como qualquer outra página web em computadores ou dispositivos móveis e mesmo assim permaneça acessível mesmo sem conexão disponível com a internet.

Este projeto visa demonstrar que um novo interpretador para a pseudolinguagem Portugal, que tem o propósito de facilitar o estudo da lógica de programação vem ao encontro como solução para muitas dificuldades atuais no âmbito de aprendizagem acadêmica. Para tal, seguiu-se a ideia de implementar este interpretador, onde haverá a preocupação em desenvolvê-lo seu protótipo em dois módulos: Interpretador e Interface Homem-Máquina além de sugerir um Sistema de Informação para o Professor. Essa ferramenta é um interpretador para a pseudolinguagem Portugal, gratuito e com uma interface adequada.

1.1 OBJETIVOS DO TRABALHO

1.1.1 Objetivo Geral

Apresentar o conceito de ferramenta computadorizada para se tornar referência em criação, edição, visualização e interpretação de algoritmos em pseudolinguagem, como alternativa às existentes. Sendo seu diferencial o acesso a partir de navegadores Web por ser em HTML5. Inclusive acessível quando uma conexão com a internet esteja indisponível.

1.1.2 Objetivos Específicos

Visando atingir o objetivo geral em sua totalidade lista-se a seguir objetivos específicos de modo sucinto:

- Desenvolver o protótipo, sendo seu código fonte aberto e disponível para que outros possam manter, ampliar e evoluir a aplicação;
- Fazê-lo compatível com sintaxe UAL e mínimo produto viável para as primeiras lições de ensino aprendizagem;
- Permitir o acesso via navegadores web modernos, sem necessidade de instalações complementares;
- Aplicar aprimoramentos introduzidos na versão 5 do HTML (*HyperText Markup Language*) para uso sem conexão com a internet.

1.2 JUSTIFICATIVA DO TRABALHO

Praticar em *software* à abstração em papel auxilia e muito no processo de ensino e aprendizagem, em algoritmos além do pensamento lógico por etapas já pode ser desenvolvido e descrito para o repasse a um computador. “O Aprendizado de algoritmos não é uma tarefa muito fácil, só se consegue através de muitos exercícios” (LOPES; GARCIA, 2002).

Para esse ensino é essencial conhecer dois conceitos principais lógica de programação e algoritmos. Primeiramente é apresentado um paralelo de situações do cotidiano com algoritmos, ao efetuar comparações de instruções passo a passo realizadas diariamente, como fritar um ovo ou trocar um pneu de automóvel. A seguir inicia-se a aplicação de linguagem estruturada em português para descrição dos algoritmos. Em alguns casos, utiliza-se de linguagens em inglês, mo por exemplo, Pascal. Há professores que preferem usar pseudo linguagem em português somente “em papel”, iniciando a codificação diretamente em linguagem C. Tem-se inclusive, também “em papel”, de analisar e interpretar os passos do algoritmo para saber quais serão os valores em memória e a saída na tela.

Simultaneamente, com a prática escrita, sem software específico, pode-se simular os mesmos algoritmos em interpretador ou compilador. Sendo em um interpretador sua construção dividida em partes, cada uma com uma função específica. Geralmente identifica-se: o analisador léxico, o analisador sintático, o analisador semântico, e o resultado de saída. Caso seja um compilador há, ao invés de resultado de saída, o gerador de código, que transforma o programa em linguagem de máquina composta de 0s e 1s (DELAMARO, 2004).

Ao utilizar um programa em navegador via Internet, fazendo com que seja desnecessário instalar e utilizar sempre o mesmo computador, objetiva-se que os exercícios serão resolvidos com maior facilidade. Também é importante que as mensagens de erro sejam claras, e em português já que torna mais acessível a leitura para quem ainda não possui facilidade com a língua inglesa.

A criação do protótipo desse sistema é o ponto principal desse trabalho, pois os objetivo é substituir *softwares* que vêm sendo utilizados atualmente por um prático, atual, acessível e com o código disponível para que possa ser corrigido e melhorado.

Nesse protótipo busca-se eliminar algumas deficiências que existem em ferramentas similares, pois essas podem fazer com que o seu uso seja desgastante e desanimador. Além de propor a inclusão de funcionalidades que poderiam estar disponíveis, as quais são encontradas por exemplo em Ambientes Integrados de Programação, comumente utilizada em outras linguagens.

Inclusive há caso em que a ferramenta inicialmente utilizada não é capaz de executar alguns algoritmos mais avançados. E isto leva a necessidade do uso de outro software na mesma disciplina até com possivelmente uma nova linguagem (por exemplo a linguagem de programação C ou C++), devido a está necessidade alguns educadores optam por esta sendo a única ferramenta utilizada, ignorando linguagem Portugol.

1.3 ESTRUTURA DO TRABALHO

No presente trabalho apresentar-se-á a forma de desenvolvimento do primeiro módulo, o interpretador, a interface com o usuário e diretrizes para a execução do módulo restante que não será executado, e que poderão ser ampliados a partir desta base de estudo e conhecimentos aplicados no presente projeto.

O restante do projeto está organizado da seguinte forma: no capítulo 2 é relatada a forma como foi idealizado o desenvolvimento da ferramenta proposta; no capítulo 3 descreve-se como o módulo interpretador está sendo desenvolvido; nos capítulos 4 e 5 são apresentadas as diretrizes que serão seguidas para os desenvolvimentos dos módulos Interface Homem-Máquina e sugestão de Sistema de Informação para o Professor, respectivamente; no capítulo 5 são apresentados as considerações finais.

2 FUNDAMENTAÇÃO TEÓRICA

2.1 ENSINO APRENDIZAGEM DE ALGORITMOS EM PORTUGOL

Portugol é uma pseudo linguagem algorítmica muito utilizada na descrição de algoritmos, destaca-se por usar comandos em português, facilitando o aprendizado da lógica de programação, e desta forma habituando o iniciante com o formalismo da programação (ESMIN, 1998).

Segundo MANZANO e OLIVEIRA (2005), o portugol pode ser classificado como uma técnica narrativa denominada de pseudocódigo, ou conhecida também por português estruturado. Baseia-se em uma PDL - *Program Design Language* (Linguagem de Projeto de Programação), sendo que sua forma original de escrita é conhecida como inglês estruturado, muito parecida com a notação da linguagem PASCAL.

Ainda segundo MANZANO e OLIVEIRA (2005), o portugol é usado como referência genérica para uma linguagem de projeto de programação, e tem por finalidade mostrar uma notação para elaboração de algoritmos, que será usada na definição, criação e desenvolvimento de uma linguagem computacional e sua documentação.

A representação de algoritmos em portugol, conforme ESMIN (1998), é rica em detalhes, como a definição dos tipos das variáveis usadas no algoritmo, encontrando muita aceitação por se assemelhar em demasia à forma em que são escritos os programas.

Comparando este processo aos fluxogramas destaca-se uma vantagem, visto que segundo MANZANO e OLIVEIRA (2005), é mais fácil escrever que desenhar (na grande maioria dos casos), e a codificação acaba se tornando uma simples transcrição de palavras chave. Corroborando ESMIN (1998) aponta o fato do Portugol ser uma linguagem simples e permitir o detalhamento dos algoritmos. A tradução de um algoritmo em Portugol para um programa computacional através de uma linguagem de programação fácil e clara, o que facilita muito o ensino/aprendizagem da própria linguagem de programação. Outro destaque, é o fato de a passagem para qualquer linguagem de programação ser quase imediata, bastando ao aluno saber as palavras reservadas da respectiva linguagem de programação.

ESMIN (1998) salienta que o inconveniente dos algoritmos é o fato de não poderem ser executados no computador, pois o iniciante precisa imaginar a sua execução, e essa tarefa não é fácil. SOUZA et al. (2000) apresenta outros problemas, tais como a influência da linguagem de programação utilizada no momento da definição do algoritmo e o fato de não haver utilização de recursos visuais, o que deixa de estimular um fator importante no aprendizado dos alunos, que é a visão.

2.1.1 O Ensino de Algoritmos

O ensino de Algoritmos, em cursos de Computação, tem por objetivo ordenar o pensamento do aluno, fazendo com que o mesmo aprenda a pensar na mesma sequência lógica utilizada pelo computador. A importância dessa disciplina será percebida mais tarde, quando o aluno iniciar o aprendizado em linguagens de programação e precisar ordenar os passos para resolução de um problema, repassando-os ao computador, sob forma de comandos (MIRANDA, 2004).

Conforme BROOKSHEAR (2000), o algoritmo é uma codificação do raciocínio necessário para resolver determinado problema, e essa capacidade torna possível a construção de máquinas com comportamento inteligente – sendo essa inteligência moldada através de software, o que torna a disciplina de algoritmos essencial ao campo de Computação. O próximo passo seria representar o algoritmo desenvolvido de uma forma apropriada ao entendimento da máquina ou do aluno, através de comandos compreensíveis e sem ambiguidade, ou seja, transportá-lo a uma linguagem de programação.

GIRAFFA et al. (2003) coloca que o objetivo da disciplina de algoritmos é o aluno desenvolver habilidades cognitivas que permitam que o mesmo aprenda a resolver problemas, utilizando-se do computador como ferramenta. Os professores, seguindo o raciocínio de GIRAFFA et al. (2003), devem se preocupar com fatores como as diferentes formas de resolução de um mesmo problema, e os diferentes estilos cognitivos de um aluno. De acordo com GARDNER (1984 apud DOMINGUES, 2003), cada ser humano possui um conjunto inato de competências intelectuais humanas chamadas de Inteligências Múltiplas. Dentre estas inteligências, existe a classificada como Lógico Matemática, chamada também de raciocínio científico ou indutivo, que é a inteligência mais desenvolvida ou potencialmente mais desenvolvida nos alunos dos cursos de Computação. GIRAFFA et al. (2003) ainda destaca que o tipo de inteligência predominante em um indivíduo não irá padronizar seu raciocínio ou a forma que o mesmo resolve problemas.

O professor da disciplina de algoritmos, de acordo com GIRAFFA et al. (2003), deve conduzir a disciplina de forma que o próprio aluno descubra seu estilo de raciocínio e forma de solução de problemas, auxiliando o aluno a modelar sua solução em forma de algoritmo. Conforme MENEZES e NOBRE (2002), o professor da disciplina de algoritmos vivencia algumas dificuldades durante o processo de aprendizagem, destacando-se:

- reconhecer as habilidades inatas de seus alunos no processo de ensino;
- apresentar diferenciadas técnicas para solução dos problemas;
- trabalhar a capacidade do aluno de buscar soluções e escolha da estrutura a ser utilizada; e
- promover a cooperação e colaboração entre os alunos.

Outro problema seria a dificuldade dos alunos em abstrair do problema o que se deseja.

Normalmente, na disciplina de algoritmos, é dada a teoria, logo após exemplos e exercícios referentes ao tema abordado. Estes exercícios normalmente são constituídos de uma parte textual, situando o aluno no problema. A grande dificuldade encontrada é abstrair deste texto o que se deseja, mais especificamente, as entradas, o processamento e a saída (MIRANDA, 2004).

CASTRO e JÚNIOR (2002) salienta que a abordagem tradicional, que é o curso baseado na programação imperativa, tem trazido sérios problemas aos alunos que nunca programaram. Esta abordagem faz com que o aluno comece a pensar um problema passo a passo, e isto não é o que é feito no dia a dia. KOLIVER et al. (2004) coloca que o despreparo da maior parte dos estudantes em questão de enfrentar uma disciplina que pretende auxiliá-los no desenvolvimento de soluções para problemas genéricos, em passos e de maneira lógica, é um dos fatores para os altos índices de desistência e reprovação.

KOLIVER et al. (2004) informa que outro fator que causa dificuldade aos professores é a heterogeneidade do público. Alguns acadêmicos possuem uma engenhosidade superior nata, o que faz com que possuam uma facilidade natural para a elaboração de soluções algorítmicas para problemas. Isso faz com que os instrutores tenham problemas em questão de definição do programa da disciplina e grau de complexidade dos problemas propostos.

Dando sequência aos problemas encontrados, normalmente os professores utilizam no ensino de algoritmos a pseudolinguagem Portugol, complementada pelo fluxograma (até parte da disciplina). Parte dos alunos sentem dificuldades em compreender estas estruturas. Conseguem analisar o problema, retirar dele as entradas, processamento e saídas solicitadas, mas no momento de transportar para a pseudolinguagem, sentem dificuldade em compreender de que forma isso deve ser feito (MIRANDA, 2004).

Outro ponto interessante para ser analisado é a forma como os estudantes chegam ao terceiro grau, conforme informa KOLIVER et al. (2004). Os calouros normalmente chegam carregando vícios oriundos de práticas mecanicistas frequentes nos primeiro e segundo graus, que acostuma o estudante a realizar aplicações de fórmulas matemáticas sem realizar qualquer tipo de análise do problema. Desta forma, o acadêmico entra na universidade sem conhecer um método geral de resolução de problemas. Isto faz com que muitos professores da disciplina possuam uma expectativa equivocada relativa ao conhecimento do aluno, esperando que o mesmo possua habilidades para análise e resolução de problemas. A ausência de disciplinas no ensino secundário que abordem a lógica também é um fator determinante para os problemas encontrados. Mesmo aplicando a lógica no dia a dia, grande parte dos estudantes possuem problemas ao aplicá-la de forma satisfatória na construção de soluções algorítmicas sem os princípios básicos que a norteiam.

Por ser uma disciplina inicial, e uma disciplina com um grau de dificuldade grande para os alunos que não conseguem se adaptar à forma de pensamento passo a passo, muitas vezes o professor sente dificuldades, durante as aulas, de avaliar qual é o problema encontrado pelos alunos. Normalmente as turmas de algoritmos são turmas grandes, e a maior parte dos alunos

não evidenciam de forma verbal os problemas encontrados. Essas dúvidas somente se tornam claras durante a aplicação de uma prova ou mesmo de exercícios válidos como nota (MIRANDA, 2004).

Pode-se também citar, como problema, a questão da correção de provas e exercícios que o professor aplica aos alunos. Como citado anteriormente, normalmente as turmas da disciplina de Algoritmos são turmas grandes, primeiro por ser uma disciplina ministrada no primeiro período - portanto, todos os alunos que ingressam através do vestibular matriculam-se nela -, pelo alto índice de reprovação. Ao corrigir provas de algoritmos, o professor deve levar em consideração que cada aluno resolve problemas de formas diferentes, portanto, um mesmo problema pode possuir diversos caminhos de resolução, e avaliar todos os problemas um a um, estabelecendo um padrão para a correção. Mas devido a grande quantidade de provas a serem corrigidas o professor, involuntariamente, poderá alterar a forma de avaliação de um aluno para outro (MIRANDA, 2004).

Apesar dos problemas citados anteriormente, a disciplina de algoritmos, baseada em Portugol e fluxograma, ainda é ministrada na grande maioria dos cursos de Computação, pois através dela o aluno consegue ter embasamento para aprender diversas linguagens de programação. A disciplina de algoritmos em si ensina o aluno a compreender um problema, definir sua resolução e aplicar esta resolução no formato da pseudo linguagem. Na grande maioria das vezes, juntamente ao ensino de Algoritmos é ensinado aos acadêmicos alguma linguagem de programação, normalmente C ou Pascal. Com o algoritmo elaborado, torna-se mais fácil o desenvolvimento do programa, pois a parte essencial, a parte lógica do problema, já está resolvida. Portanto, o aluno deve somente transferir a resolução para a linguagem de programação (MIRANDA, 2004).

2.2 O ENSINO DA LINGUAGEM DE PROGRAMAÇÃO PORTUGOL

O ensino da lógica de programação geralmente é tratado nas primeiras fases dos cursos de informática, onde os alunos iniciantes aprendem a desenvolver o raciocínio lógico para então escrever algoritmos para solução de problemas. O Portugol é uma pseudo linguagem algorítmica muito utilizada na descrição de algoritmos, destaca-se pelo uso de comandos em português, o que facilita o aprendizado da lógica de programação, habituando o iniciante com o formalismo de programação.

Apesar de todas essas vantagens, o Portugol apresenta o inconveniente dos algoritmos não poderem ser executados no computador. Dessa forma, o iniciante precisa imaginar a sua execução, o que não é uma tarefa tão fácil para quem está começando.

A lógica para programação consiste em aprender a pensar na mesma sequência em que o computador executa as tarefas, aprende-se a imaginar como as ações serão executadas partindo-se do estudo de um problema até chegar a construção de um algoritmo (solução). Considere com exemplo o seguinte problema:

“Expressar o resultado da soma de dois valores.”

É comum, para uma pessoa pensar em algo assim:

“Pegar os dois valores, somar e dar o resultado.”

Após ter adquirido uma certa experiência, a mesma pessoa pode, de forma automática, converter tal pensamento em instruções sem necessidade de especificar detalhadamente os processos que estão implícitos nesta pequena rotina (ver tabela 1):

Tabela 1 – Ações e Operações

Ações	Operações
PEGAR	Receber os dois valores numéricos e armazená-los
SOMAR	Executar a instrução de soma e armazenar o resultado
DAR O RESULTADO	Mostrar o resultado, armazená-lo para o uso posterior ou para ser visualizado em outra oportunidade

Fonte: Farrer (1989)

Quanto maior o domínio da lógica de programação, mais fácil será detalhar as tarefas envolvidas na solução do problema proposto e mais eficiente será o algoritmo criado, porém, para um iniciante construir um algoritmo que permita a um computador executar a tarefa proposta não é tão simples.

Segundo (FARRER et al., 1989, p. 17): “Algoritmo é a descrição de um conjunto de comandos que, obedecidos, resultam numa sucessão finita de ações”. A tabela 1, mostra duas leituras de um algoritmos, uma bastante genérica (ações) e outro um pouco mais detalhada (operações), porém, apesar de apresentar um bom nível de detalhamento ainda não pode ser considerado um algoritmo para o computador. Em Portugol, o mesmo algoritmo pode ser escrito da seguinte forma:

Algoritmo exemplo ;

Var v1 ,v2 , v3 : **inteiro** ;

Início

leia (v1) ;

leia (v2) ;

v3 <- v1 + v2 ;

escreva (v3) ;

Fim.

A representação dos algoritmos em Portugol, conhecido também como pseudocódigo, é muito utilizada, segundo (SALIBA, 1992, p. 6): “Esta forma de representação de algoritmo é rica em detalhes, como a definição dos tipos das variáveis usadas no algoritmo e, por assemelhar-se bastante à forma em que os programas são escritos, encontra muita aceitação.”

A tradução de um algoritmo escrito em Portugol para um programa de computador numa linguagem de programação é muito fácil e clara, facilitando assim o ensino e aprendizado da linguagem de programação. O Portugol é a forma mais utilizada para escrever algoritmos por ser uma linguagem simples e permitir o detalhamento dos algoritmos.

O ensino do Portugol, hoje, em muitos casos é feito de maneira manual (utilizando folhas de papel), o que não estimula os alunos em aprender e exercitar o desenvolvimento de algoritmos.

2.2.1 A Ferramenta Portugol/Plus

O Objetivo principal do sistema Portugol/Plus é fornecer uma ferramenta de apoio ao ensino da lógica de programação baseado no Portugol, sem reduzir o estudo teórico. Com esta ferramenta pretende-se proporcionar uma forma de estimular os alunos a praticar e exercitar o desenvolvimento de algoritmos em Portugol.

O ambiente foi desenvolvido para operar em microcomputador do tipo PC (*Personal Computer*), sob o sistema operacional DOS (*Disk Operating System*), e divide-se em duas partes principais:

- Editor de algoritmos (padrão ASCII);
- Compilador.

1. Editor de Algoritmos:



Figura 1 – Visão Geral do Ambiente

O editor de algoritmos, permite a digitação e a manipulação dos arquivos que contém os algoritmos utilizando uma interface dirigida por menus. Numa visão geral (figura 1) vê-se os diversos menus (Arquivo, Editar, Localizar, Compilar, Janela e Opções):

- O menu arquivos permite, como o nome diz, a manutenção de arquivos, com itens como criar, salvar e salvar como (salvar com outro nome), contando também com ações extras, tais como:
 - Mudar o dir que serve para alterar o diretório (área de trabalho) em que o ambiente pesquisa os arquivos para edição.
 - Dos shell que serve para se obter uma sessão de linha de comando do sistema operacional sem abandonar o ambiente.
 - Sair para encerrar a execução do ambiente.
- O menu editar permite executar ações sobre os arquivos em edição (abertos), estas ações compreendem copiar, mudar e/ou apagar partes dos arquivos em edição.
- O menu localizar ajuda na busca e/ou substituição de trechos escritos no arquivo atual em uso (corrente).
- O menu janela controla o modo como são visualizados todos os arquivos abertos (em edição) em um mesmo instante.
- O menu opções controla valores configuráveis pelo usuário dentro do ambiente como cores e mouse.
- O menu opções controla valores configuráveis pelo usuário dentro do ambiente como cores e mouse.

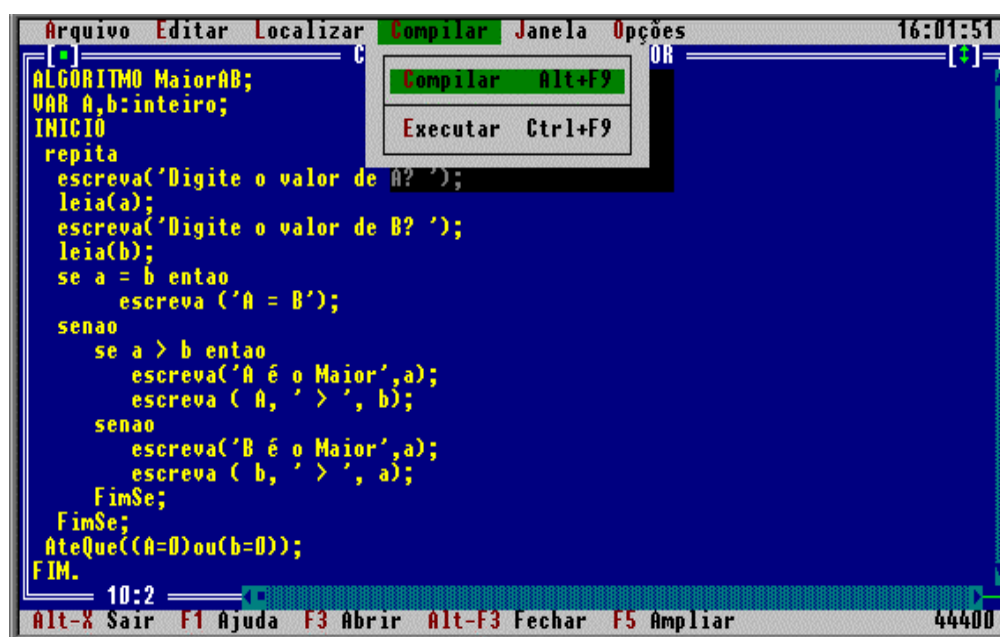


Figura 2 – Visão do Menu Compilar

- O menu compilar (figura 2), serve para tornar possível a execução dos algoritmos criados. Após editado e salvo, usa-se este menu para a realização da compilação do algoritmo, depois de corrigidos os erros (caso seja encontrado algum), pode-se executar o algoritmo para ver o seu comportamento, se o resultado obtido é realmente o desejado. Podendo-se também, o que é muito importante para o aprendizado, partir para sucessivos refinamentos (melhorias) nas soluções encontradas, sempre tendo-se a possibilidade de visualizar como as alterações feitas afetam o resultado pretendido (a nova solução pode vir a ser melhor, pior ou equivalente).

2. Compilação:

O conhecimento sobre a construção de compiladores, segundo (AHO et al., 1995; NETO, 1987), remonta do início da década de 50, não se sabendo precisamente o seu início. A construção de compiladores foi motivada pela complexidade de programar em computadores na época, quando usava-se código de máquina ou *Assembler*, o que além de difícil, forçava a execução do programa no equipamento para o qual o mesmo havia sido escrito, pois em outros o código precisava ser reescrito, o que resultava em versões do mesmo programa para cada tipo de equipamento.

Os compiladores são programas que recebem com entrada um programa fonte, analisam-no o conteúdo e o contexto, caso encontrem erros informam ao usuário, do contrário, convertem o programa fonte para um programa objeto geralmente em linguagem da máquina.

O compilador trabalha sobre uma gramática, que é um conjunto de palavras (instruções) limitado por ele e reconhecidas como comandos, que tem regras fixas e não conflitantes de uso, o que significa que uma instrução tem apenas um significado para um compilador, não havendo ambiguidade.

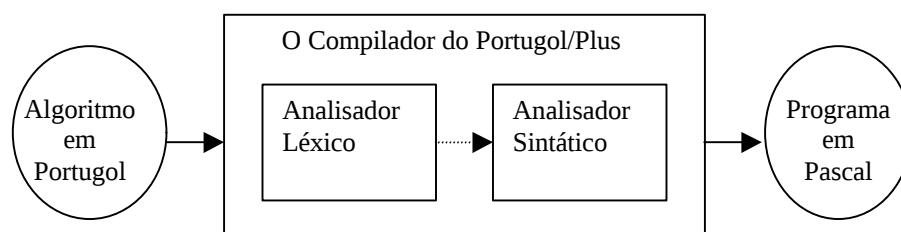


Figura 3 – O Compilador

Durante o processo de compilação como mostrado na figura 3, o compilador do ambiente Portugol/Plus faz uma verificação da sintaxe das instruções usadas no algoritmo, caso não sejam encontrados erros, um programa objeto é gerado na linguagem de programação Pascal (FARRER et al., 1995). Através de um compilador Pascal executa-se o programa objeto.

A programação de computador, não é algo muito complexo de ser aprendida, no entanto, para o seu aprendizado é necessário dominar a lógica de programação. O ensino da lógica depara-se hoje com uma problemática, que é a falta de experimentar na prática o que é estudado na teoria. A falta de prática desestimula os alunos iniciantes a aprofundar no conteúdo a ser aprendido (ESMIN, 1998).

Visando a melhoria do ensino da lógica foi desenvolvido o Portugol/Plus, que é um sistema que auxilia na tarefa de ensinar e estudar a lógica de programação. Ele estimula o aluno a praticar e exercer o desenvolvimento de algoritmos, facilitando assim o trabalho do professor e também ajudando o aluno a ter um domínio melhor sobre o assunto (ESMIN, 1998).

O principal público alvo são os alunos iniciantes dos cursos de Informática, onde se espera, com a utilização desta ferramenta, um aumento no domínio do conteúdo da lógica (ESMIN, 1998).

2.3 PROCESSAMENTO DE LINGUAGEM NATURAL

BARR e FEIGENBAUM (1986) colocam que o caminho mais comum para que as pessoas se comuniquem é falando ou escrevendo através de linguagem natural, seja em inglês, francês, português ou chinês. As linguagens de computadores possuem um formato mais rígido, para que possam ser convertidas em uma sequência de instruções de computador. O estudo de processamento de linguagem natural procura fazer com que os computadores possam entender a linguagem natural humana, tornando-se mais fáceis de utilizar.

REFERÊNCIAS

AHO, A. V.; SETHI, R.; ULLMAN, J. D. **Compiladores: princípios, técnicas e ferramentas**. Tradução de Daniel de Ariosto Pinto, Traduzido de: Compilers, principles, techniques and tools. Rio de Janeiro: LTC, 1995.

BARR, A.; FEIGENBAUM, E. A. **The Handbook of Artificial Intelligence**. [S.l.]: AddisonWesley Publishing Company, INC, 1986. v. 1.

BROOKSHEAR, J. G. **Ciência da Computação: Uma Visão Abrangente**. Porto Alegre: Bookman, 2000.

CASTRO, T. H. C.; JÚNIOR, A. N. C. Utilizando programação funcional em disciplinas introdutórias de computação. In: **X Workshop sobre Educação em Computação - XXII Congresso Nacional da Sociedade Brasileira de Computação na Universidade Federal de Santa Catarina - UFSC**. Florianópolis: Anais do XXII Congresso da Sociedade Brasileira de Computação, 2002.

DELAMARO, M. E. **Como Construir um Compilador Utilizando Ferramentas Java**. São Paulo: Novatec, 2004. Disponível em: <<http://conteudo.icmc.usp.br/pessoas/delamaro/SlidesCompiladores/CompiladoresFinal.pdf>>.

DOMINGUES, M. J. C. d. S. Estimulando as inteligências múltiplas através da realidade virtual. Universidade Federal de Santa Catarina - UFSC, Florianópolis, 2003.

ESMIN, A. A. A. Portugol/plus : Uma ferramenta de apoio ao ensino da lógica de programação baseado no portugol. In: **IV Congresso Ibero-americano de Informática Educativa**. [s.n.], 1998. Disponível em: <http://www.niee.ufrgs.br/eventos/RIBIE/1998/pdf/com_pos_dem/118.pdf>.

FARRER, H.; BECKER, C. G.; FARIA, E. C. **Pascal estruturado**. 2. ed. Rio de Janeiro: Guanabara, 1995.

FARRER, H. et al. **Algoritmos estruturados**. Rio de Janeiro: LTC, 1989.

GARDNER, H. **Estruturas da mente: a Teoria das Múltiplas Inteligências**. Porto Alegre: Artes Médicas, 1984.

GIRAFFA, L. M. M.; MARCZAK, S. S.; ALMEIDA, G. C. O ensino de algoritmos e programação mediado por um ambiente na web. Anais do XXIII Congresso da Sociedade Brasileira de Computação. Campinas, SP: UNICAMP, Campinas, IV, p. 19–28, 2003. Disponível em: <<http://www.lbd.dcc.ufmg.br/colecoes/wei/2003/003.pdf>>.

KOLIVER, C.; DORNELES, R. V.; CASA, M. E. Das (muitas) dúvidas e (poucas) certezas do ensino de algoritmos. In: **XXIV Congresso Nacional da Sociedade Brasileira de Computação na Universidade Federal da Bahia**. Salvador: Anais do XXIV Congresso da Sociedade Brasileira de Computação, 2004.

LOPES, A.; GARCIA, G. **Introdução à programação**. Rio de Janeiro: Campus-Elsevier, 2002.

MANZANO, J. A. N. G.; OLIVEIRA, J. F. d. **Algoritmos - Lógica para Desenvolvimento de Programação de Computadores**. 4. ed. São Paulo: Editora Érica Ltda, 2005.

MENEZES, C. S.; NOBRE, I. A. M. Um ambiente cooperativo para apoio a cursos de introdução a programação. In: **X Workshop sobre Educação em Computação - XXII Congresso Nacional da Sociedade Brasileira de Computação na Universidade Federal de Santa Catarina - UFSC**. Florianópolis: Anais do XXII Congresso da Sociedade Brasileira de Computação, 2002.

MIRANDA, E. M. **Ferramenta de Apoio à Correção de Algoritmos propostos utilizando Processamento de Linguagem Natural**. Florianópolis: Universidade Federal de Santa Catarina - UFSC, 2004.

NETO, J. J. **Compiladores: princípios, técnicas e ferramentas**. Rio de Janeiro: LTC, 1987.

SALIBA, W. L. C. **Técnicas de programação: uma abordagem estruturada**. São Paulo: Markin, 1992.

SOUZA, E. M. S. et al. Um novo enfoque para o processo ensino-aprendizagem de algoritmos em cursos de graduação. Anais do I Simpósio Catarinense de Computação, Itajaí, v. 1, p. 80–88, ago 2000.