Instituto de Ciências Matemáticas de São Carlos

ISSN - 0103-2569

MANUAL DO USUÁRIO

PROTÓTIPO DE UM EDITOR GENÉRICO SENSÍVEL À SINTAXE
(Versão para Windows V3.1)

CARLOS ROBERTO VALÊNCIO
MAURICIO SGAVIOLI ROCCHI
CLAUDIA MARTINS SOUZA
EMERSON YUKIHISA KOMI
CAETANO TRAINA JÚNIOR

Nº 41

RELATÓRIOS TÉCNICOS DO ICMSC

São Carlos
Jun./1996

UNIVERSIDADE DE SÃO PAULO

Instituto de Ciências Matemáticas de São Carlos

Departamento de Ciências de Computação e Estatística

MANUAL DO USUÁRIO

PROTÓTIPO DE UM EDITOR GENÉRICO SENSÍVEL À SINTAXE

(VERSÃO PARA WINDOWS V3.1)

Prof. Carlos Roberto Valêncio

Mauricio Sgavioli Rocchi

Cláudia Martins Souza

Emerson Yukihisa Komi

Prof. Dr. Caetano Traina Júnior

USP - São Carlos

Estado de São Paulo

Junho - 1996

Índice

1.0 - Introdução

A primeira versão do Protótipo do Editor Genérico Sensível à Sintaxe - PEGSS foi construída no ICMSC - USP, São Carlos como trabalho de mestrado [VAL_93], sendo implementado em C e compilado através do AZTEC Versão 4.1.

O PEGSS é denominado genérico, porque permite a edição de programas em qualquer linguagem entre as várias linguagens suportadas pelo Editor. Além disso, uma vez editado o programa na linguagem escolhida, é possível obter-se o mesmo programa representado em qualquer uma das outras linguagens, dentro do que seja possível para a tradução de um programa escrito de uma linguagem para outra. Diz-se sensível à sintaxe porque o editor tem o conhecimento da construção sintática da linguagem utilizada na edição do programa, possibilitando ao final, um programa livres de erros léxicos e sintáticos. Além das características descritas acima, o Editor é dirigido por estruturas, pois a edição é feita basicamente através da requisição de gabaritos que são alocados em uma composição hierárquica de objetos sintáticos, ao invés de uma seqüência de caracteres. A construção dessa estrutura durante o processo de edição, permite representar o programa de modo estruturado, e no processo de interação com o usuário, apresentar as estruturas sintáticas do mesmo nível uma abaixo da outra, e as dos níveis inferiores aparecem obedecendo a uma endentação.

Os programas editados são armazenados em uma base de dados, segundo a concepção lógica de uma construção sintática genérica para as linguagens suportadas pelo PEGSS, denominada de Descrição da Linguagem Abstrata (DLA). Através da DLA é possível obter-se a representação de um programa em qualquer uma das linguagens, pois um programa é armazenado em modo único para todas elas. Essa descrição é armazenada também na base de dados, de maneira que linguagens que tenham apenas construções sintáticas equivalentes à qualquer uma das suportadas podem ser incluídas para serem suportadas, bastando para isso incluir sua descrição na base.

O opção pelas linguagens PASCAL, C e FORTRAN deve-se basicamente as construções sintáticas dessas linguagens serem em muitos casos parecidas, possibilitando uma representação genérica das suas estruturas sintáticas. Estas características visam fazer do PEGSS uma ferramenta de suporte à fase de codificação, dentro do ciclo de desenvolvimento de software, possibilitando que programas sintaticamente corretos sejam escritos sob o seu controle, e que módulos escritos em uma linguagem possam ser utilizados para composição de outros, em outra linguagem.

A atual versão do Protótipo do Editor Genérico Sensível à Sintaxe pode ser compilada no ambiente Borland C++ V3.1 for Windows. O padrão Windows escolhido melhorou a interface do Editor com o usuário. Para a instalação do PEGSS é necessária uma máquina 386 ou melhor, com o ambiente Windows instalado

Os tópicos a seguir apresentam as opções oferecidas pelo PEGSS e a sua funcionalidade.

Sabendo que as linguagens Pascal, C e FORTRAN pertencem ao mesmo paradigma, os exemplos dos tópicos a seguir são na maioria em Pascal. O PEGSS permite que o mesmo programa seja apresentado nas três linguagem a partir de uma mesma base de dados, portando, os exemplos podem ser melhor analisados em quaisquer das linguagens suportadas pelo Editor.

2.0 Instalação

Criando-se um diretório PEGSS copia-se todos os arquivos contidos no meio de instalação.

Digite *instala* para que os arquivos sejam descompactados.

Após a descompactação haverá um arquivo chamado PEGSS.EXE, execute este arquivo no Windows para trabalhar com o Editor.

O kit de instalação pode ser obtido por FTP ou WEB, através dos endereços :

- ftp://icmsc.sc.usp.br/pub/gdb/pegss/instala
- http://icmsc.sc.usp.br/~gdb/pegss/editorss

3.0 - Protótipo de um Editor Genérico Sensível à Sintaxe - PEGSS

Como foi dito anteriormente, a versão atual do PEGSS funciona em ambiente Windows. Portanto o mouse, os ícones e as teclas de atalho podem ser utilizados.

O menu principal possui as opções **Arquivo**, **Níveis**, **Utilitários** e **Help,** cujas funções são:

- Arquivo: **Novo, Abrir, Salvar, Fechar e Sair;**
- Níveis: **Mostrar até nível...** e **Voltar edição;**
- Utilitários : **Editar Documento** e **Mostrar Identificadores;**
- Help: **Sobre;**

A Figura 1 mostra a tela principal do programa.

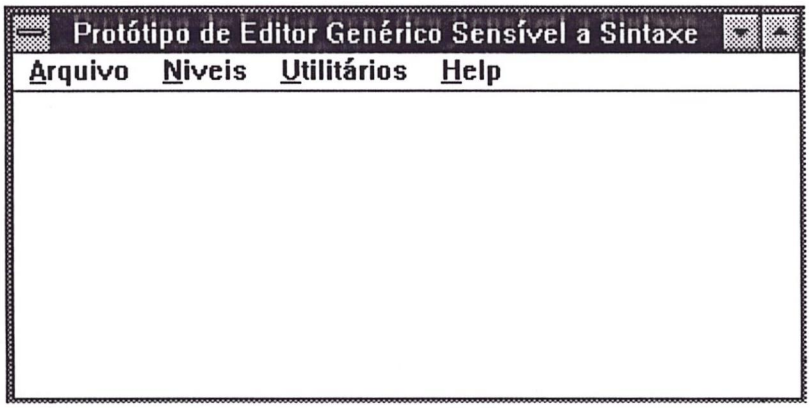

Figura 1 : Menu Principal do Editor

O funcionamento das opções é explicado mais detalhadamente nos tópicos a seguir.

4.0 - Manipulação de Arquivos

Escolhendo-se a opção **Arquivo** pelo menu ou pressionando as tecla <ALT>+<N> simultaneamente, aparecerão as novas opções : **Novo, Abrir, Salvar, Fechar e Sair.** Estas opções são explicadas a seguir.

4.1 - Criando um Novo programa

Através da opção Arquivo do menu permite-se a criação de um novo arquivo de programa utilizando o PEGSS.

Escolhendo-se a opção **Novo**, aparecerá na tela uma janela com as opções de linguagens suportadas pelo Editor: PASCAL, C, FORTRAN como mostra a Figura 2.

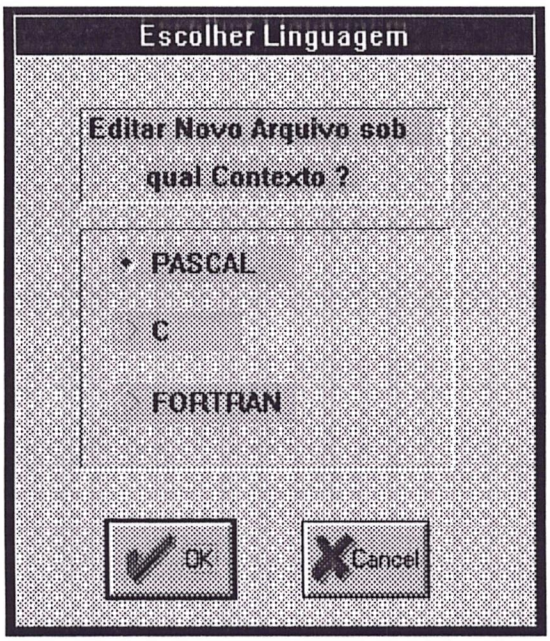

Figura 2 : Escolha da Linguagem

Após a escolha da linguagem é solicitado o nome do novo arquivo a ser criado. Se este arquivo já foi criado em seções anteriores, será emitida uma mensagem pedindo a confirmação da sua recuperação. Caso contrário, o PEGSS pedirá um outro novo nome de arquivo para começar a edição.

Se a criação do arquivo foi efetuada corretamente e a linguagem escolhida for PASCAL, o Editor mostrará o gabarito inicial para esta linguagem, como mostrado na Figura 3. Após a criação do novo arquivo, a opção **Novo** do menu **Arquivo** ficará desabilitada, impedindo de ser escolhida até que o programa atual seja fechado, selecionando a opção **Fechar** ou pressionando a tecla <ESC>.

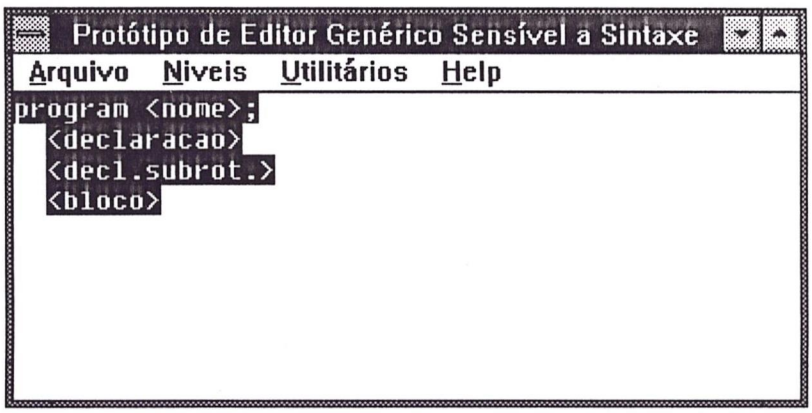

Figura 3: Gabarito inicial para um programa em Pascal

Neste momento, a partir do gabarito inicial proposto pelo Editor, inicia-se a construção do programa, onde o programador poderá inserir, através dos lugares de inserções ("placeholder"), novas estruturas sintáticas.

A seleção do lugar de inserção desejado é efetuada pela utilização das teclas de movimentação do cursor: setas para cima, para baixo, para os lados, etc. Uma vez escolhido o "placeholder" adequado, pressiona-se a tecla <RETURN> para que seja possível continuar o processo de inserção de gabaritos. Para mais detalhes, ver seção 7.0.

4.2 - Recuperando programas já editados.

A opção **Abrir** trata da abertura de um de programa já editado armazenado na base de dados.

Novamente, é necessário escolher a linguagem em que se deseja recuperar o arquivo.

Devido ao modo único de representação na base, o programa poderá ser recuperado em qualquer das linguagens suportadas pelo Editor. Escolhida a linguagem, aparecerá na tela uma janela com os arquivos já existentes (Figura 4). O arquivo selecionado é recuperado, podendo assim, continuar sua edição na linguagem corrente. Esta opção fica inacessível ao usuário quando um arquivo está sendo editado. Após o usuário fechar o programa corrente, esta opção volta a ficar disponível.

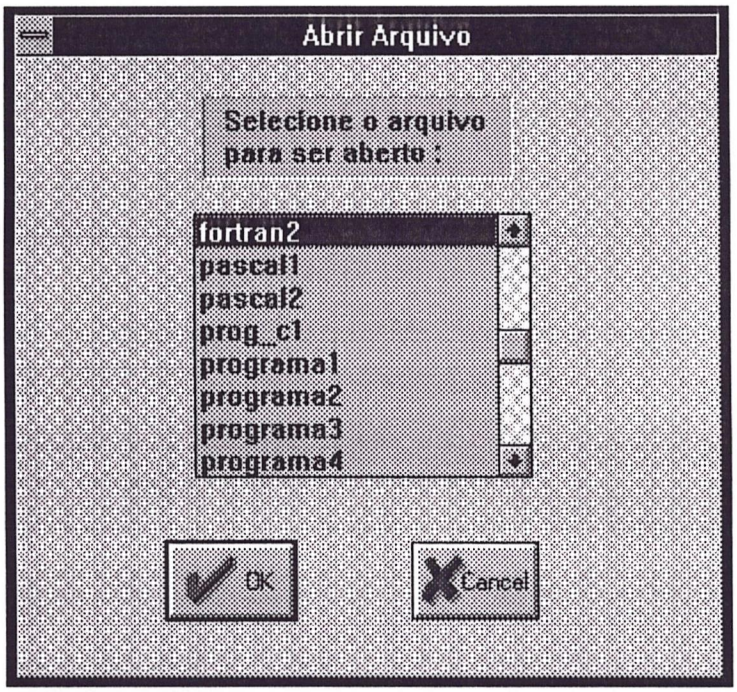

Figura 4 : Lista de arquivos de programas

4.3 Armazenando o programa editado

Inicialmente a opção **Salvar** fica inativa, pois não existe arquivo sendo editado. Após o início da edição, esta opção fica acessível ao usuário. Nesta versão do software, a opção **Salvar** não está totalmente implementada, pois a partir do momento que o usuário iniciar a edição de um novo arquivo, ele já vai estar armazenado na base de dados. Portanto, a qualquer momento que o usuário sair do programa, ele já vai estar salvo na base de dados. Em termos de base de dados, esta opção corresponde a tratar todo o processo de edição como uma única transação, que será confirmada com a operação de salvamento.

4.4 Encerrando a Edição de um programa

A opção **Fechar** fecha o arquivo editado, deixando acessível as opções **Novo** e **Abrir** do menu. Esta ação pode ser executada pressionando a tecla <ESC>.

A partir deste momento o usuário pode trabalhar com outro arquivo.

4.5 Saindo do Ambiente do Editor

A opção **Sair** fecha o programa ativo, e gera uma mensagem perguntando se o usuário deseja sair do PEGSS, ou ainda deseja editar mais programas.

5.0 - Apresentação do programa em níveis de detalhe

A opção **Níveis** mostra o funcionamento de uma das ferramentas implementadas oferecidas pelo PEGSS. A ferramenta permite que se visualize o programa editado em diferentes níveis de detalhe. As Figuras 5, 6, 7 e 8 mostradas a seguir ilustram melhor o funcionamento da ferramenta.

Antes de se escolher a opção **Níveis** é necessário que um programa esteja carregado, ou seja, não é possível mostrar um programa em diferentes níveis de detalhe se ele não estiver sendo editado. Escolhida a opção **Níveis**, aparecerão duas novas opções: **Mostrar até nível...** e **Voltar edição.**

5.1. Escolhendo o nível de detalhe

Quando é escolhida a opção **Mostrar até nível...**, aparecerá uma janela onde será pedido o número do nível, ou seja, o valor que definirá o grau de detalhe que será apresentado o programa.

Seja o programa "prog1" da Figura 5.

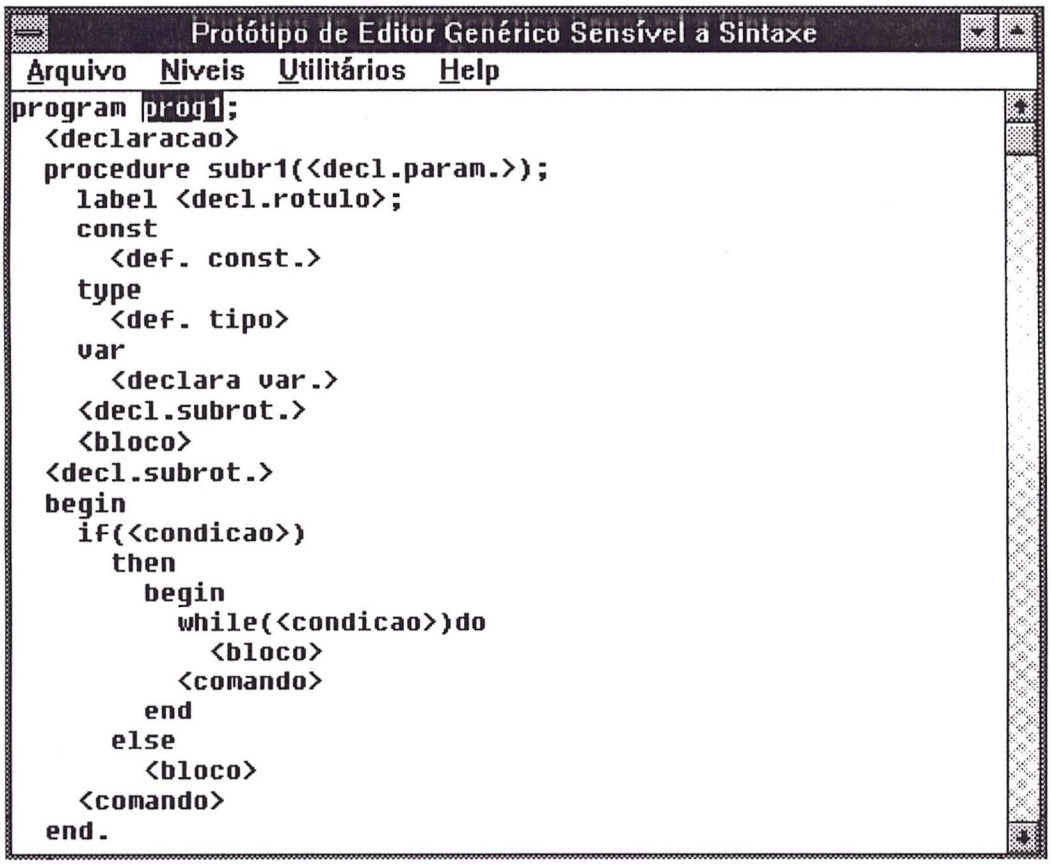

Figura 5: Programa 'prog1'

Caso o usuário escolha o nível 2, o programa será apresentado conforme os detalhes mostrados na Figura 6.

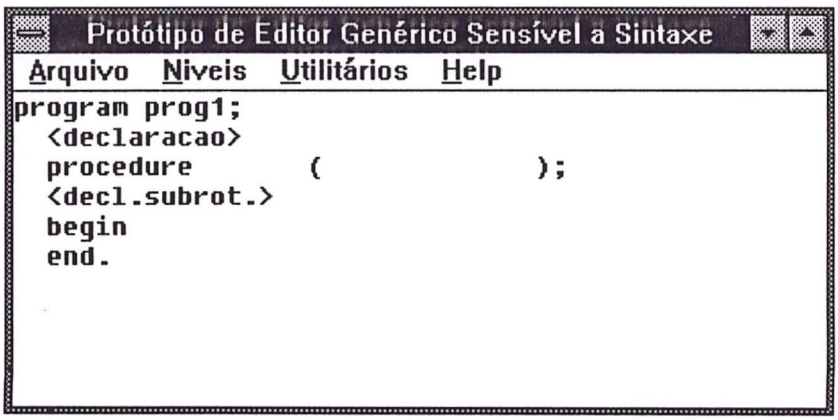

Figura 6: Programa 'prog1' com nível 2 de detalhe

Para apresentar mais detalhes sobre os procedimentos e sobre o bloco principal do programa, o usuário escolheria um nível maior. Caso o usuário desejasse observar até o nível 4, o programa ficaria como mostrada na Figura 7.

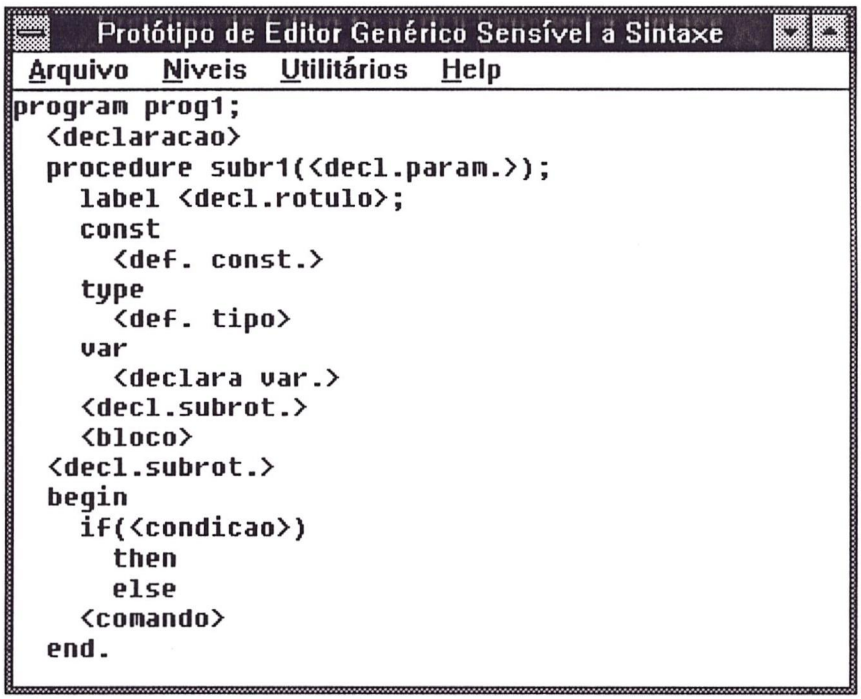

Figura 7: Programa 'prog1' com nível 4 de detalhe

Para expandir ainda mais os detalhes do programa, observe a Figura 8, onde o nível escolhido é o 6.

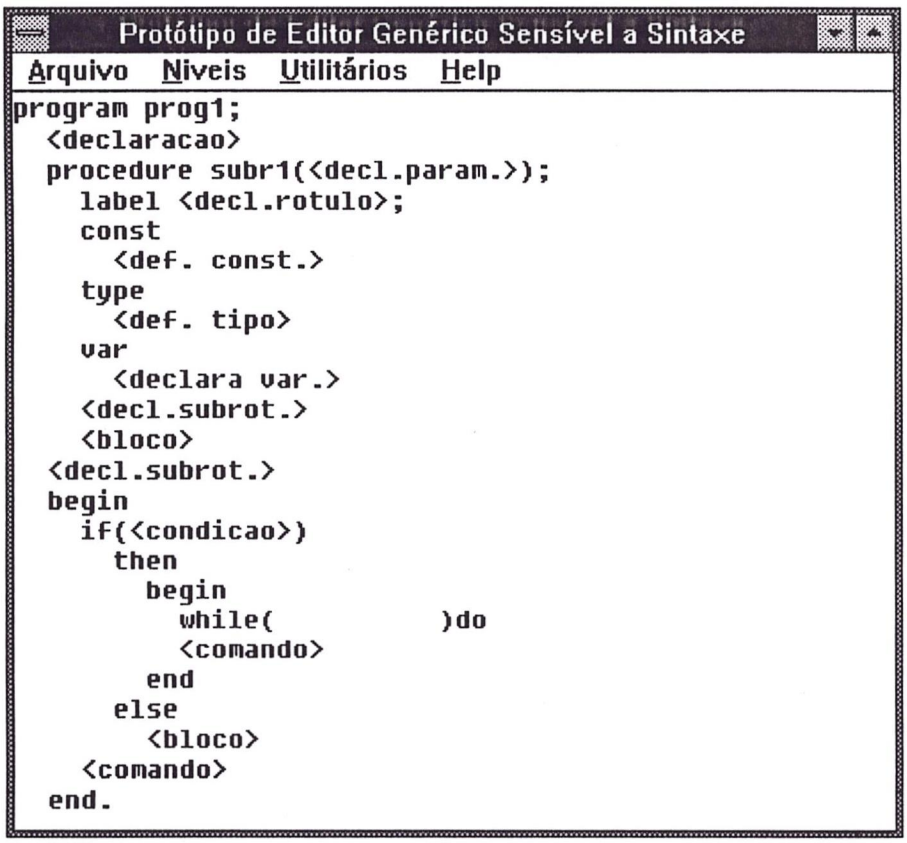

Figura 8: Programa 'prog1' com nível 6 de detalhe

Note na figura 8, que a expressão do comando "*while*" não é mostrado, pois toda expressão é armazenada em um nível posterior. Caso o usuário escolha o nível 7, neste exemplo, a expressão é mostrada.

As figuras mostram, respectivamente, o programa com nível de detalhe 2, 4, 6..., observando que os detalhes do programa aumentam a medida que cresce o nível. Quando o programa for mostrado por completo significa que o programa atingiu o nível máximo de detalhe. Mesmo que se escolha um nível maior a tela permanecerá inalterado.

5.2. Voltando ao processo de Edição

A opção **Voltar edição** faz com que o programa seja editável novamente e poderá ser utilizada somente quando se escolhe a opção **Mostrar até nível....** Utilizando-se a ferramenta de verificação de níveis, não é permitida a edição do programa. Isto só torna-se possível quando o usuário escolher a opção **Voltar edição**, ou teclar <ESC>.

6.0 - Outras ferramentas

Através da opção **Utilitários** são oferecidos um conjunto de ferramentas que atuam sobre o programa editado

6.1 - Apresentação da documentação gerada

Selecionando-se a opção **Editar Documento...**, o Editor pesquisará na base de dados o nome do arquivo, o nome do programa, os identificadores globais, os nomes de todas as subrotinas definidas e seus respectivos identificadores, ou seja, apresentará um conjunto de informações pertinentes ao programa selecionado. Considerando o programa da Figura 9, estas informações são mostradas numa caixa de edição (ver Figura 10), onde o usuário poderá acrescentar qualquer comentário.

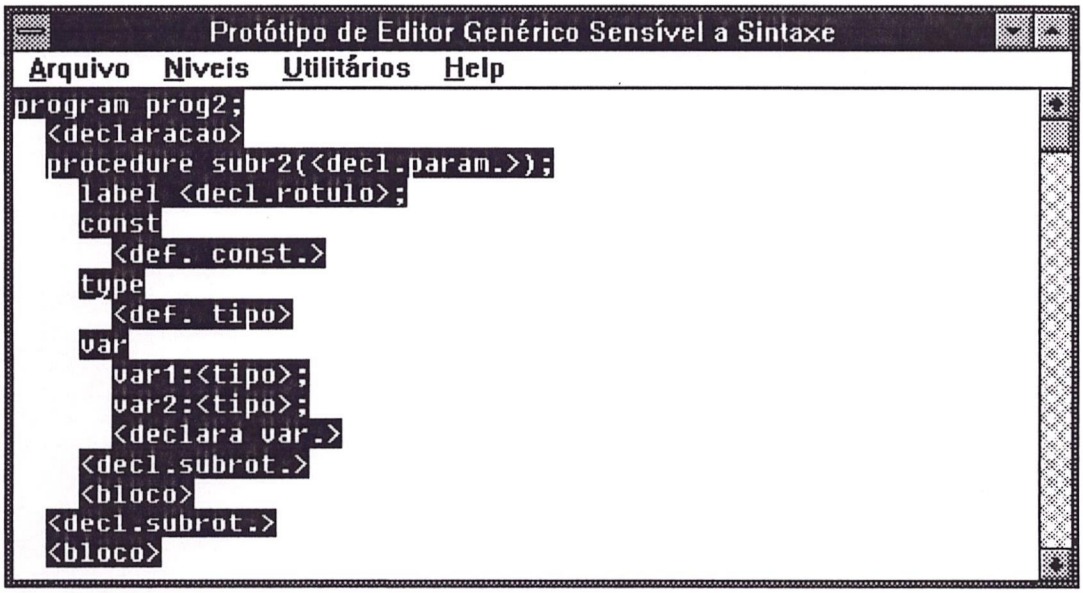

Figura 9: Programa 'prog2'

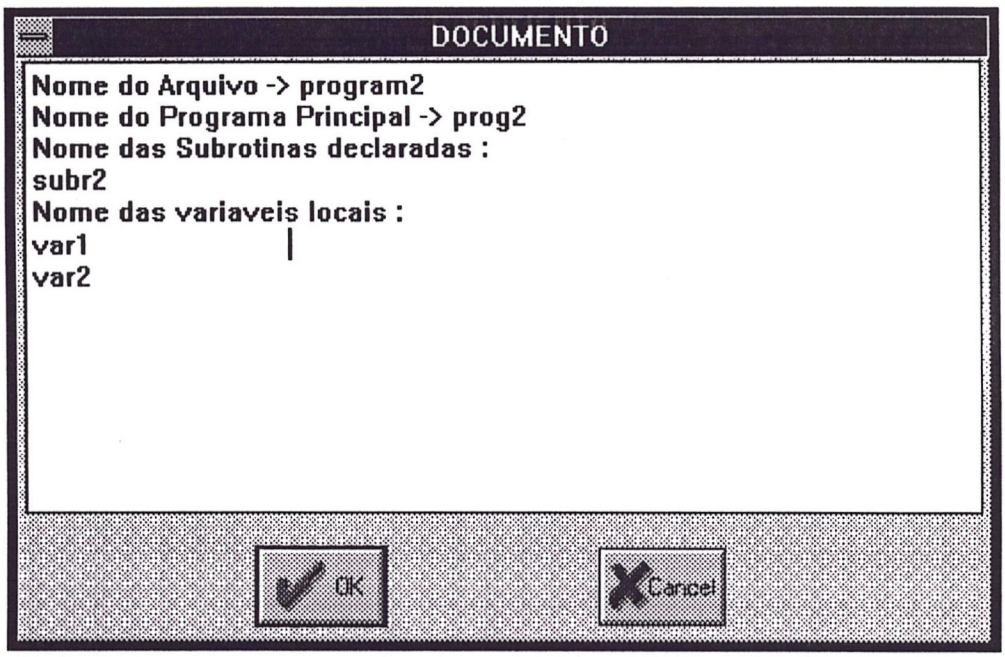

Figura 10: Documentação gerada pelo Editor sobre o programa 'prog2'

Após a edição dos comentários feita pelo usuário, será criado um arquivo texto, contendo toda a documentação do programa. Esta caixa pode ser aberta em qualquer momento durante a edição do programa, possibilitando assim, a criação de vários arquivos de apoio a documentação.

6.2 - Apresentação dos Identificadores utilizados em uma subrotina

A opção **Mostrar Identificadores** tem como objetivo mostrar ao usuário todos os identificadores que foram declarados numa determinada subrotina. Ao escolher esta opção, aparecerá uma caixa para a inserção do nome da subrotina. Caso este nome não tenha sido definido, uma mensagem de aviso é gerada. Após a pesquisa na base de dados, aparecerá uma caixa onde é mostrado o nome das subrotinas e todos os identificadores declarados (ver Figura 11).

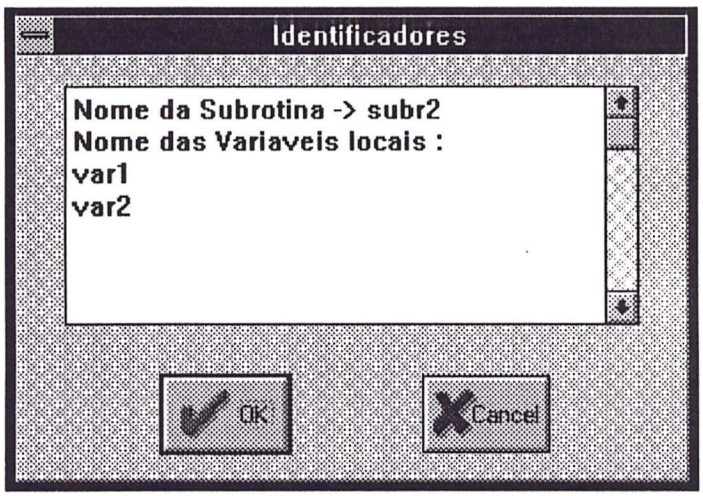

Figura 11: Lista dos identificadores definidos para a subrotina 'subr2'

7.0 - Edição

As expressões e comandos de atribuição no PEGSS são introduzidos digitando-se caracter por caracter numa caixa de diálogo, enquanto as demais estruturas da linguagem são oferecidas como gabaritos, através das escolhas em caixas de diálogos que contém os comandos suportados pelo Editor.

Um gabarito representa uma estrutura sintática da linguagem composto por palavras-chave e lugares para as possíveis introduções de outros gabaritos ou expressões. Erros léxicos são evitados, pois não é permitido ao programador alterar as palavras-chave que compõem o gabarito, além da verificação léxica imediata das expressões inseridas. Por exemplo, um gabarito de uma estrutura de desvio condicional utilizada pelo PEGSS é:

if (<condição>)
  then
    <bloco>
  else
    <bloco>

onde as palavras **<condição>** e **<bloco>** formam os lugares para as possíveis inserções, e o "if", "then" e "else" são palavras-chave.

Após a escolha da linguagem e introdução do nome do arquivo, o processo de criação de um programa é iniciado a partir de um primeiro gabarito, oferecido pelo Editor. Considerando que a linguagem corrente é o Pascal, o gabarito inicial é mostrado na Figura 12.

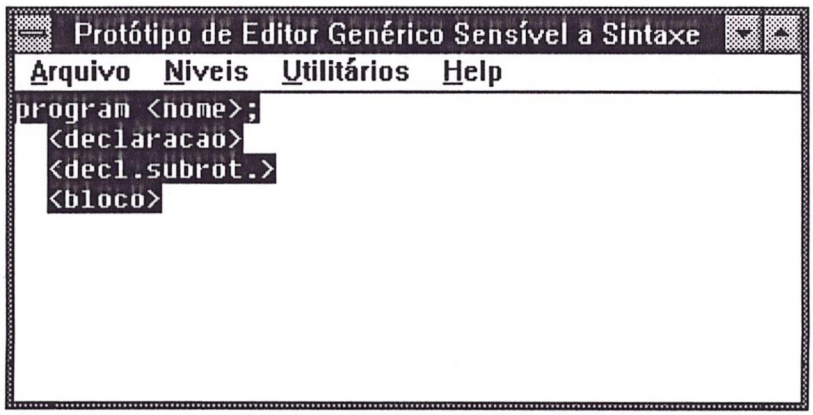

Figura 12: Estrutura inicial de um programa em Pascal

Este primeiro gabarito aparece com a palavra-chave e os lugares de inserção em destaque ("highlight") indicando que esta é a estrutura corrente no momento. A tecla <RETURN> é utilizada para as inserções de novos gabaritos ou expressões e, se pressionada nesta posição, não causará efeito algum, pois tais inserções devem ser efetuadas quando o elemento corrente for um lugar de inserção. Consequentemente, as palavras-chave não são acessíveis ao usuário, que não tem possibilidade de alterá-las.

A movimentação ("navegação") sobre o gabarito é proporcionada através da utilização das teclas seta para cima < ↑ >, seta para baixo < ↓ > , seta para a esquerda < ← >, seta para a direita < → > , <HOME> e <END>. Se a tecla seta para a direita for pressionada, a palavra que ficará em destaque será **<nome>** e, pressionando a tecla seta para baixo, a palavra em destaque será **<declaração>**. Repetindo o último passo duas vezes o lugar de inserção posicionado será o **<bloco>**. Esta seqüência de movimentações sobre o gabarito é mostrado abaixo, através das Figuras 13, 14, 15, 16 respectivamente :

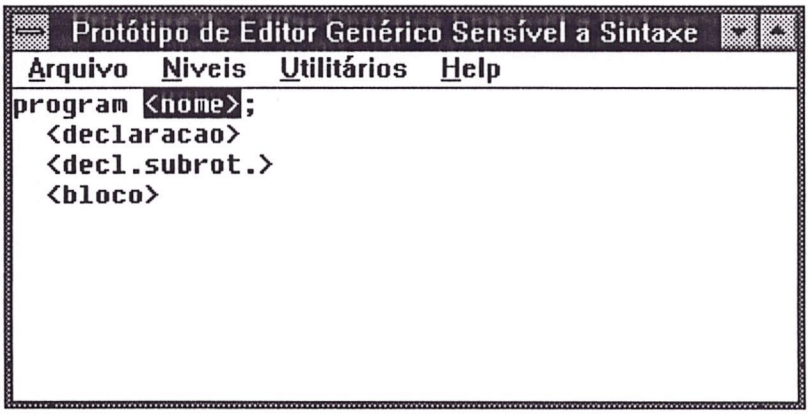

Figura 13: Lugar para inserção <nome> em destaque

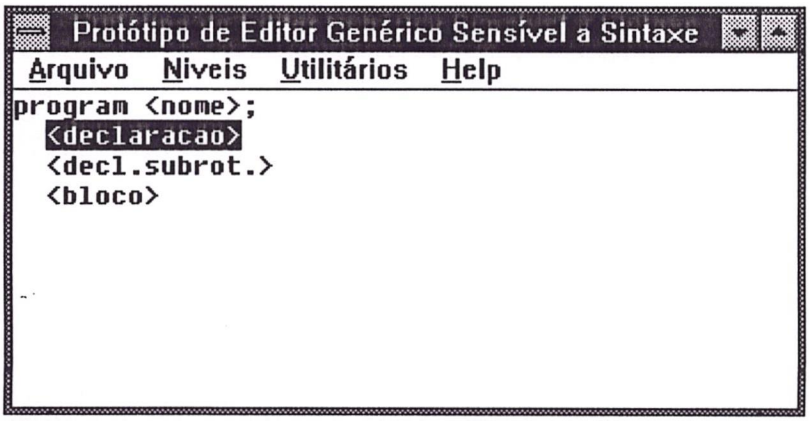

Figura 14: Lugar para inserção <declaração> em destaque

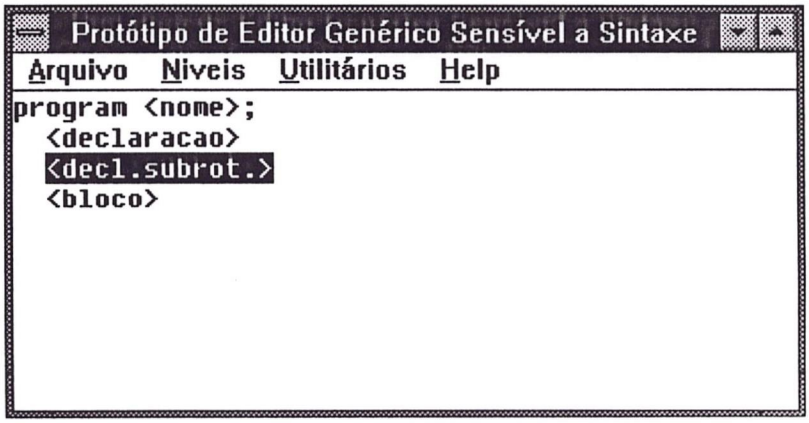

Figura 15: Lugar para inserção <decl. subrot.> em destaque

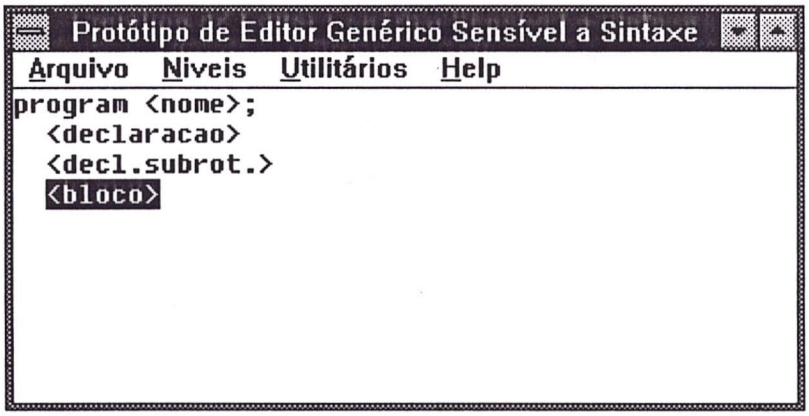

Figura 16: Lugar para inserção <bloco> em destaque

Ao pressionar a tecla <RETURN>, um novo gabarito será apresentado em seu lugar, contendo :

begin

<comando>

end.

Após posicionar no lugar de inserção **<comando>** (ver Figura 17) e pressionar a tecla <RETURN>, é apresentada uma lista de opções de escolha entre os comandos com ou sem rótulos. A escolha de comandos sem rótulos proporciona a apresentação de um outro menu de opções contendo as possíveis estruturas sintáticas a serem introduzidos neste ponto, como é mostrado na Figura 18.

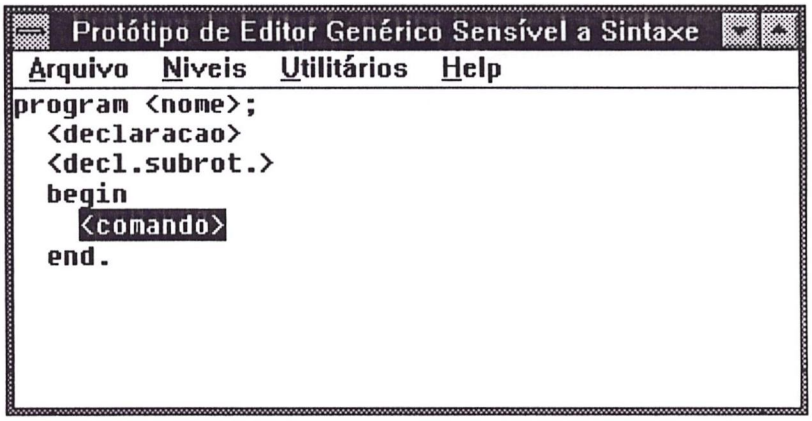

Figura 17: Posição <comando> permite a inserção de novas estruturas

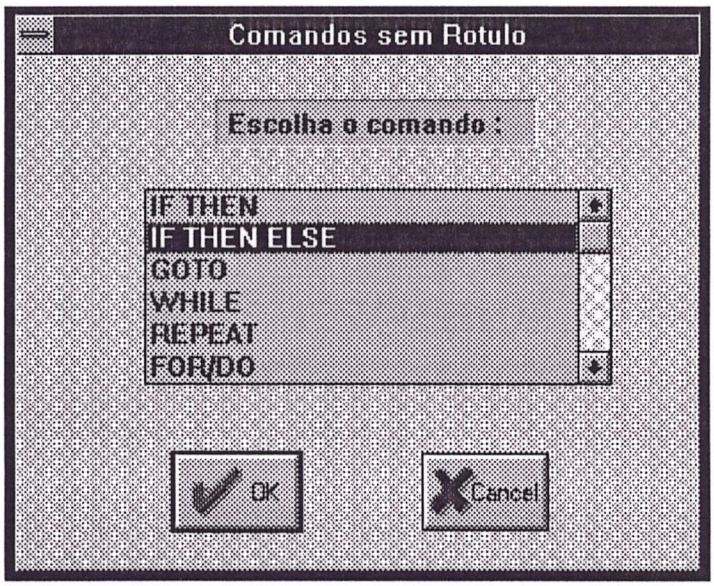

Figura 18: Estruturas disponíveis para a inserção através de <comando>

Escolhida a estrutura, um gabarito, representando-a, será colocado no lugar de **<comando>**. Se a estrutura escolhida for *IF/THEN/ELSE*, o programa ficará conforme é mostrado pela Figura 19 :

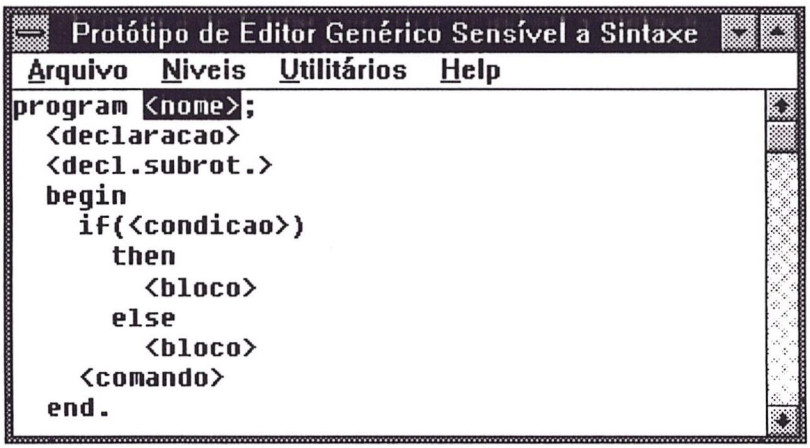

Figura 19: Programa após a inserção da estrutura *if/then/else*

Sempre que um lugar de inserção é substituído por um gabarito, um outro lugar será oferecido pelo Editor, para que a construção do programa possa prosseguir através de inserções de mais estruturas sintáticas, pertencentes à mesma classe sintática do lugar substituído. No exemplo acima, o lugar de inserção **<comando>** foi gerado pelo Editor, após a inserção da estrutura selecionada.

Os lugares de inserção como **<nome>** e **<condição>** permitem que identificadores e expressões sejam, respectivamente, introduzidos. Este processo é efetuado através de uma caixa de dialogo que permite a introdução dos caracteres que os compõem. Após sua descrição, estes serão substituídos nos seus respectivos lugares de inserção. Na introdução de um identificador já utilizado no programa é emitida uma mensagem avisando o programador de tal fato. Posicionando no lugar de inserção **<nome>** (ver Figura 19), e pressionando <ENTER>, aparecerá uma caixa para a inserção do nome do programa (ver Figura 20). Após a introdução do nome, e posicionando em **<condição>**, aparecerá a caixa de dialogo(ver Figura 21) para o usuário colocar a expressão desejada.

Figura 20: Inserção de um identificador

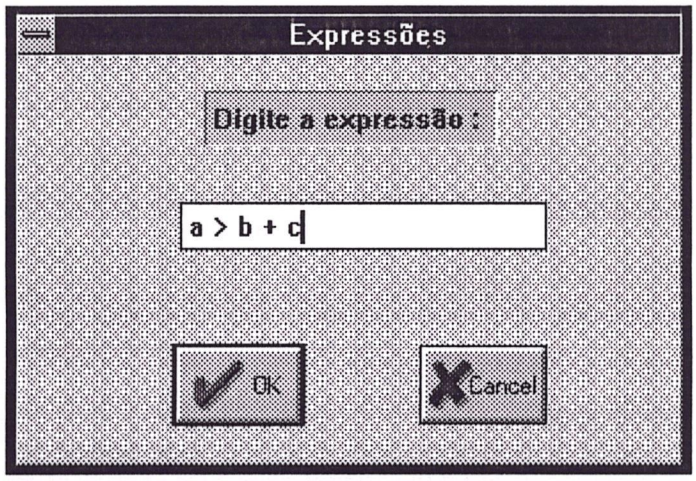

Figura 21: Inserção de uma expressão

Após estas inserções o programa editado ficará conforme a Figura 22.

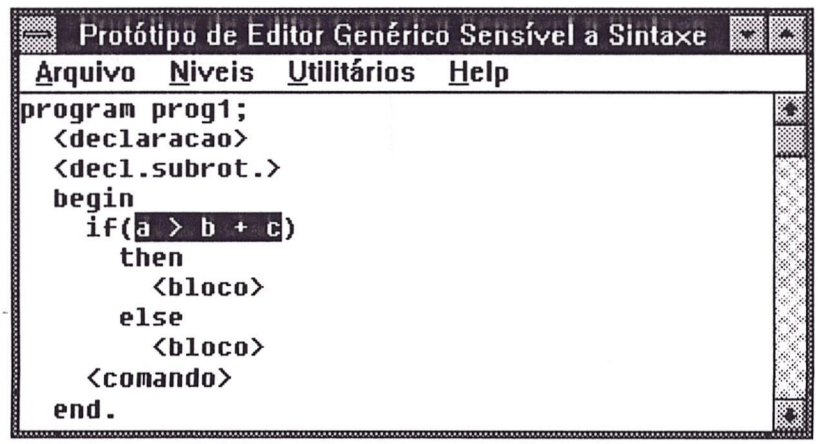

Figura 22: Apresentação da expressão digitada no programa 'prog1'

Deste modo o programa é escrito através do processo "top-down", onde as introduções são dirigidas pelos lugares de inserção definindo a correção sintática do mesmo. Somente a análise sintática das expressões ainda não é efetuada nesta versão do PEGSS, ocorrendo apenas a sua verificação léxica.

8.0 - Expressões

Toda expressão é introduzida no PEGSS pelo usuário. Isso é efetuado, digitando-se os caracteres que a descrevem, o que difere do modo de inserção das outras estruturas sintáticas em um programa.

Com o cursor posicionado em qualquer lugar de inserção de expressões e pressionando-se a tecla <RETURN>, será permitido que o programador possa inserir os caracteres que descrevem a expressão, através de uma janela de edição de expressões (ver Figura 21) . Em seguida, é efetuada a análise léxica. Após a analise léxica é iniciado o processo de preparação da expressão para o seu armazenamento. Caso existam erros léxicos, a expressão é desprezada e uma nova poderá ser introduzida.

A expressão digitada sofre uma preparação para seu posterior armazenamento na base de dados, transformando-a em uma expressão genérica. A expressão é assim armazenada possibilitando a sua respectiva representação nas outras linguagens do Editor.

9.0 Referências Bibliográficas

[ROCC_95] Mauricio Sgavioli Rocchi; Claudia Martins Souza; EmersonYukihisa Komi; Caetano Traina Júnior e Carlos Roberto Valêncio, **"Ferramentas de Suporte a Edição Sensível à Sintaxe: Exibição de Programas em Níveis de Detalhes e um Gerador de Documentos de Programas"**, Anais do 3° Simpósio de Iniciação Científica da Universidade de São Paulo, São Carlos - SP, pp. 232, Dezembro de 95.

[SOU_95] Claudia Martins Souza; Mauricio Sgavioli Rocchi; Emerson Yukihisa Komi; Caetano Traina Júnior e Carlos Roberto Valêncio, **"Verificação Automática e Tipos e Declarações de Variáveis em um Editor Sensível à Sintaxe"**, Anais do 3° Simpósio de Iniciação Científica da Universidade de São Paulo, São Carlos - SP, pp. 231, Dezembro de 95.

[TRA_88] Caetano Traina Júnior, e Jan Frans Willem Slaets, **"Um Modelo de Representação de Objetos"**, Anais do 3° Simpósio Brasileiro de Banco de Dados, Recife, pp. 227-242, Março de 1988.

[TRA_91] Caetano Traina Júnior, **"GEO: Um Sistema de Gerenciamento de Base de Dados Orientado a Objetos - Estado Atual de Desenvolvimento e Implementação"**, Anais do 6° Simpósio Brasileiro de Banco de Dados, Manaus, pp. 193-207, Maio de 1991.

[TRA_93] Caetano Traina Júnior e Carlos Roberto Valêncio, **"Documentação Técnica do Editor Genérico Sensível a Sintaxe"**, Relatórios Técnicos do ICMSC - USP - São Carlos, Novembro de 1993.

[VAL_93] Carlos Roberto Valêncio e Caetano Traina Júnior, **" Protótipo de um Editor Genérico Sensível à Sintaxe"**, Anais do 3° Simpósio de Pós-Graduação e Pesquisa em Ciências Exatas e Engenharias, Atibaia-SP, Novembro de 1993.

[VAL_93] Carlos Roberto Valêncio, **"Um Editor Genérico Sensível a Sintaxe Armazenada em uma Base de Dados"**, Dissertação de Mestrado - ICMSC - USP- São Carlos, Fevereiro de 1993.