DEPARTAMENTO DE CIÊNCIA DA COMPUTAÇÃO

Relatório Técnico

RT-MAC-8808

Um Sistema Simples para Documentação Semi Automática de Programas

V. W. SETZER

UNIVERSIDADE DE SÃO PAULO
INSTITUTO DE MATEMÁTICA E ESTATÍSTICA
SÃO PAULO - BRASIL

1. INTRODUÇÃO.

A documentação de programas é um dos aspectos cruciais do desenvolvimento de sistemas computacionais em qualquer ambiente. Desde o desenvolvimento de "software" básico, até o de "software aplicativo" para processamentos administrativos, a atividade de programação usando alguma linguagem de programação requer um esforço específico na documentação de sistemas e programas. Esse fato independe da linguagem: essa necessidade existe desde a programação em linguagem de montagem ("assembler") até o uso de linguagens denominadas de "4ª geraçâo".

Temos verificado continuamente que as documentações de sistemas e programas em geral simplesmente inexistem ou são inadequadas e desatualizadas. As principais consequências desses fatos são:

C1. A manutenção de programas é muito cara e demorada.
C2. A fase de testes de programas demanda um tempo exagerado e sobrecarrega os analistas e programadores.
C3. Não há segurança e confiança de que os programas estão corretos.
C4. O autor de um programa torna-se o "pai" do mesmo, pois ninguém mais consegue compreendê-lo e mantê-lo.
C5. Não há meios para se verificar a documentação, isto é, se ela realmente reflete a estrutura do programa.
C6. A gerência não consegue saber se um programa está bem documentado.
C7. O projeto, a programação e a documentação devem ser feitos pelas mesmas pessoas.

Neste artigo apresentamos um sistema extremamente simples de documentação de programas através de comentários feitos na sintaxe (de comentários) da própria linguagem de programação do programa. Um programa também simples (apresentado aqui como exemplo da documentação em sua versão em Pascal para o VAX) é empregado para gerar automaticamente tres níveis de documentação. Delineamos também um programa para gerar automaticamente a documentação das alteraçoes efetuadas em um programa, feitas em um editor de textos qualquer. Veremos como esse sistema também pode forçar metodologias "top-down" de programação (refinamentos sucessivos), como permite introduzir e controlar padrões de documentação e como diminui as consequências apresentadas acima.

2. NÍVEIS DE DOCUMENTAÇÃO.

Em nosso sistema empregamos três níveis de documentação; ele pode ser facilmente estendido para conter mais níveis, se for conveniente. Os níveis contêm o seguinte:

Nível 1: Informações gerais sobre os programas e procedimentos.
Nível2: Toda a documentação de nível 1, mais estruturas conceituais de dados e de programas e rotinas.

Nível 3: Todas as documentações de níveis 1 e 2, mais o programa propriamente dito, com documentação de seus detalhes.

Note-se o encaixamento dos conteúdos de cada nível em relação aos níveis anteriores.

3. ESPECIFICAÇÕES DOS NÍVEIS.

A documentação de nível 1 é inserida no programa sob forma de comentários ocupando linhas completas (isto é, não deve haver nada, nessas linhas, além desses comentários), que denominaremos de comentários de nível 1. Eles são identificados colocando-se o dígito '1' após o símbolo de início de comentário da linguagem, que deve ser inserido na posição mais à esquerda da linha (coluna 1), como por exemplo '(*1', '{1' ou '*1' (dependendo da linguagem). O símbolo de fim de comentário deve ser colocado nas últimas posições da linha.

A documentação de nível 2, como vimos, contém a de nível 1, produzida pelos comentários de nível 1, e mais os comentários de nível 2. Estes são linhas de comentários de formato idêtico ao dos comentários de nível 1, excetuando-se o dígito '2' colocado no lugar do '1'.

A documentação de nível 3 contém as duas anteriores, e mais todos os comandos do programa e comentários de nível 3. Estes são inseridos no programa em qualquer posição das linhas, usando-se simplesmente os símbolos de comentários da linguagem empregada. Evidentemente, não devem confundir-se com os comentários de nível 1 e 2; portanto, não devem conter os dígitos '1' ou '2' imediatamente após o símbolo de início de comentário.

4. LISTADOR DA DOCUMENTAÇÃO.

O "listador" da documentação é um programa que solicita do usuário a informação do nível de documentação desejado, e em seguida varre sequencialmente o arquivo com o programa, produzindo uma listagem contendo o seguinte:

a) Cabeçalho em cada página, contendo o nome do programa, o nível de documentação, a data e a hora da listagem e o número da página.
b) Se o nível desejado é 3, a listagem conterá todas as linhas do programa, incluindo todos os comentários, precedida cada uma por um número de linha. Esse número é o número de ordem de cada linha do programa, incluindo todos os comentários dos vários níveis.
c) Se o nível desejado for 1 ou 2, a listagem conterá as linhas de comentário extraídas do programa, conforme o nível desejado, pre-cedidas do número de linha correspondente à listagem do nível 3.

Por meio dos números de linhas pode-se estabelecer rapidamemte uma correspondência visual entre as linhas das listagens dos vários níveis de documentação.

5. EXEMPLO. DETALHES DAS DOCUMENTAÇÕES.

No apêndice apresentamos como exemplo as listagens dos três níveis de documentação para o programa COMENT.PAS. Esse programa é justamente um listador para programas escritos em Pascal, COBOL ou ZIM. Ele foi escrito em Pascal para o VAX. A partir desse exemplo especificaremos aquilo que, até o presente momento, consideramos como interessante constar da documentação de cada nível.

5.1. Nível 1.

Note-se que ao programa principal e a cada procedimento corresponde um trecho da documentação deste nível. Esta deve ser totalmente **conceitual**, isto é, deve ser voltada para o problema (ou "mundo real", se for o caso). Assim, num sistema de conferição de notas fiscais colocaríamos "verificação do ICM" e não "verificação do campo NFICM do registro R2". Além disso, deve ser totalmente auto-contida, isto é, deve ser independente das documentações de níveis 2 e 3, a menos dos números de linhas. Denominamos nessa documentação de "entradas" e "saídas" os dados transportados de e para unidades externas, respectivamente. Denominamos de "importações" e "exportações" os dados que são globais a ou são parâmetros de procedimentos; no primeiro caso acrescentamos a palavra "globais".

Note-se, ainda, a enumeração dos nomes dos procedimentos chamados ("rotinas chamadas") e quais procedimentos chamam uma certa rotina ("chamado por").

5.2. Nível 2.

Observe-se que a documentação deste nível contém a de nível 1; distinguimos os comentários de níveis 1 e 2 por meio da impressão do dígito que constava do início do comentário correspondente.

Como no nível 1, os comentários de nível 2 devem ser totalmente conceituais. Além disso, devem ser totalmente independentes da documentação de nível 3, a menos dos números de linhas.

No exemplo, não colocamos as estruturas dos dados, pois elas são demasiado elementares; se existissem, deveriam ser independentes da estrutura "fisica". Por exemplo, deveriam comentar o mais conceitualmente possível o conteúdo dos arquivos como conjuntos ou tabelas (visão relacional) em relação com o problema ou "mundo real".

A estrutura dos programas é dada em um "Português estruturado". Chamamos a atenção para o fato de que procuramos evitar em nosso "Português estruturado" o uso de palavras reservadas das linguagens de programação usuais pois cremos que a

descrição de algoritmos deve libertar-se das amarras estruturais e linguísticas impostas por essas linguagens. Assim, não precedemos as condições com "se..." para evitar propositalmente a conotação com o 'if'. De fato, essas condições estão formuladas como perguntas com interrogação, e podem gerar tanto estruturas como 'if' quanto 'case'. Infelizmente, não encontramos uma alternativa melhor do que "caso contrário" para o "senão". É importante nesse nível tratar os dados como conjuntos e elementos de conjuntos. Por exemplo, uma certa malha de repetição ("loop") poderia ser especificada como "processa cada linha da nota fiscal".

Atenção especial deve ser dada para o alinhamento vertical: é ele que dá a estrutura de composição das ações. A regra empregada é a seguinte: se uma linha m começa na coluna i, a próxima linha n a ser executada sequencialmente depois de m, independentemente da composição desta, deve ser a próxima linha que também começa na coluna i; entre m e n não deve ocorrer nenhuma linha com algum caractere em colunas de índice menor ou igual a i.

Note-se a correspondência da numeração das documentações de nível 1 e de nível 2.

5.3. Nível 3.

Note-se que a documentação deste nível contém todo o programa, com os comentários de níveis 1, 2 e 3, exatamente como ocorrem no arquivo.

Observe-se que todas as variáveis do programa foram documentadas através de comentários de nível 3. Esse é o nível correto para isso, pois variáveis não têm nada a ver com a descrição conceitual dos níveis superiores. Além disso, foram colocados alguns comentários de nível 3 para comandos do programa.

Procuramos seguir a mesma regra de alinhamento vertical exposta para os comentários de nível 2. Infelizmente, o resultado não foi exatamente o desejado, mas é suficiente para ilustrar essa nossa única regra; a existência de uma só regra de alinhamento, ao contrário de maneiras usuais de alinhar comandos de programas, que contêm regras específicas para certos comandos, tem a intenção de uniformizar e simplificar essa disposição, independentemente da linguagem.

Finalmente, note-se como os comandos do programa aparecem inseridos nos comentários de nível 2; cada um destes está documentando conceitualmente a ação executada pelos comandos que o seguem e que precedem o próximo comentário daquele nível.

6. DOCUMENTAÇÃO DAS ALTERAÇÕES.

Se se deseja documentar o histórico de todas as alterações efetuadas em um programa, a simples listagem das várias versões do mesmo não é prática, pois é relativamente difícil deduzir quais foram as modificações introduzidas empregando-se uma comparação visual entre as listagens de duas versões consecutivas. Muitos programadores empregam um programa especial de comparação de arquivos, que salienta as diferenças. Essa solução não é

satisfatória em casos de muitas alterações. Vejamos a solução que nosso sistema oferece para esse problema.

Afim de documentar as alterações projetamos um programa de listagem de alterações LA, que funciona da seguinte maneira:

a) O usuário ativa LA e comunica a este que deseja iniciar alterações; fornece ainda o nome do arquivo, por exemplo A, onde está seu programa P e o nome do editor de textos E que deseja utilizar para editar as alterações.
b) LA abre um novo arquivo A1 e copia P em A1, numerando suas linhas.
c) LA passa o controle ao editor E dando A1 como origem do texto.
d) O usuário altera A1 empregando E, **sem alterar nenhuma numeração de linhas**; ele pode alterar o conteúdo de linhas de P, inserir neste novas linhas (sem colocar a numeração destas) ou eliminar linhas (eliminando também sua numeração). O usuário deve também inserir linhas de comentário especiais no início de A1, usando um código de comentário especial como '/#' a partir da coluna 1; essas linhas servem para documentar-se a razão de se ter feito as alterações. O programa alterado, que denominaremos de P1, é então gravado em A1.
e) O usuário ativa LA novamente e comunica a este que deseja emitir as listagens de alterações, fornecendo os nomes de A e de A1.
f) LA varre A numerando internamente as linhas de P. Em paralelo, lê A1 comparando as suas linhas com as de mesmo número de A. Se houver alguma diferença de conteúdo entre duas linhas de mesmo número, ambas são gravadas em um arquivo de saída S, a de A precedida da mensagem "original" e a de A1 de "alterada". Se houver alguma linha de A com um número de linha que não ocorre em A1, ela é gravada em S precedida de "eliminada". Se ocorrer uma sequência de linhas de A1 sem numeração, estas são gravadas em S precedidas da mensagem "linhas inseridas", da linha de A correspondente à linha numerada de A1 imediatamente anterior à sequência, e seguidas pela linha de A imediatamente seguinte. Todas as linhas de A1 com código de comentário especial ('/#') são também inseridas em S.
g) À medida que LA varre A1 no passo anterior, vai regravando A1 retirando as numerações de linhas e eliminando as linhas com código de comentário especial ('/#').
h) Ao terminar a varredura de A e de A1, LA executa uma rotina de listagem semelhante ao listador, que le S e produz a impressão das alterações em todos os níveis de documentação; cada nível é precedido dos comentários especiais ('/#').

Como resultado desses procedimentos permanece o programa P em A, P1 é criado em A1 e obtém-se a listagem das alterações efetuadas na passagem de P para P1, divididas nos vários níveis de documentação, com as numerações das linhas de P para referência com o programa original e precedida dos comentários especiais que descrevem os motivos das alterações. O arquivo S pode ser novamente

empregado para obter-se novas cópias das alterações, bastando para isso ativar LA e informar que se deseja executar esse passo.

Note-se que esse sistema de documentar as alterações serve tanto para registrar-se as alterações de manutenção, como também para documentar os passos de desenvolvimento de um programa em um método por refinamentos sucessivos.

Existem vários sistemas de documentação de alterações, denominados de "version control systems"; uma descrição de vários desses sistemas pode ser encontrada em [1]. Em particular, o sistema DIFF originário do UNIX e transposto para o TURBO-C é muito empregado no IBM-PC. Cremos que nossa proposta é superior a esses produtos pois está voltada especificamente para o sistema de documentação aqui descrito, além de adaptar-se a qualquer editor de textos. Note-se que a numeração das linhas feitas pelo nosso sistema resolve todos os problemas de comparação entre as linhas sem prejudicar o usuário, pelo contrário, está dentro do esquema da documentação produzida pelo listador.

6. RESULTADOS.

Com o método aqui exposto pode-se obter os seguintes resultados:

R1. Documentação de um programa, do seu desenvolvimento e de sua manutenção em vários níveis de abstração.

R2. Programação "top-down". Deve-se desenvolver um programa começando-se pelos comentários de nível 1, a seguir inserir os de nível 2, e finalmente o programa propriamente dito e os comentários de nível 3.

R3. Geração automática de documentação.

R4. Aceleração do desenvolvimento. A documentação em vários níveis auxilia a programação e diminui o tempo dedicado aos testes.

R5. Aceleração da manutenção. Para se fazer manutenção de um programa, pode-se localizar o ponto de alteração (ou de erro) "navegando-se" pelos níveis decrescentes de abstração da documentação. Os números de linhas podem ser empregados para uma rápida movimentação de um nível para outro.

R6. Atualização permanente da documentação. Qualquer alteração do programa deve redundar em alteração nos comentários de níveis 3, 2 e mesmo 1 se necessário. Os listadores de alterações e de documentação produzirão a documentação das alterações efetuadas e da nova versão da documentação do programa.

R7. Programação encadeada. Várias pessoas podem participar do desenvolvimento de um programa, particionando-se a equipe conforme os tres níveis de abstração. Isso garante a qualidade das documentações de níveis 1 e 2.

R8. Controle da programação "top down". A listagem das alterações correspondendo ao desenvolvimento "top down", contendo as várias versões do refinamento sucessivo, possibilita à gerência verificar se a metodologia está sendo corretamente seguida.

R9. Controle da documentação. A gerência pode controlar a documentação dos programas, já que é relativamente fácil seguir

as documentações de níveis 1 e 2. A partir destas, não deve ser difícil seguir o programa distribuído entre elas, e seus comentários de detalhe. A documentação das alterações permite controlar a manutenção das documentações dos níveis mais altos.

R10. Aumento da portabilidade. Alterações de programas devidas a mudanças de ambiente de processamento podem ser feitas com muito maior rapidez, pois devem afetar muito pouco os níveis 1 e 2 de documentação. Em particular, comentários de nível 3 podem ser inseridos para salientar as dependências de trechos do programa em relação ao ambiente.

R11. Documentação independente dos detalhes de programação. A documentação de nível 1 pode ser empregada por pessoas que desejam conhecer a funcionalidade dos programas e das rotinas sem se importar com detalhes de programação.

7. CONCLUSÕES.

Apresentamos aqui um método extremamente siples de produzir e manter a documentação de programas. Estamos cônscios de que esta não é uma área promissora, pois a programação está desaparecendo devido à introdução de geradores universais de aplicações para processamento de dados administrativos, que em alguns casos como o sistema LDT geram automaticamente programas e documentação, sendo esta de altíssimo nível [2]. No entanto, sobram ainda todos os desenvolvimentos de "software" básico, quando não são usadas técnicas formais como VDM (ver, como exemplo dessa técnica, [3]), e a grande maioria dos sistemas desenvolvidos em linguagens de programação algorítmicas e de "4a. geração". Outros esforços tem sido produzidos no sentido de se introduzir comentários significativos nos programas, como o sistema WEB de Knuth [4]. No entanto, trata-se de sistema muito mais complexo, abrangendo inclusive a parte de asserções para prova de corretude (para uma extensa referência a respeito, ver [5]). Essas asserções podem evidentemente ser introduzidas em nosso método como comentários de nível 2. O mesmo se passa com comentários de tipo formal como os propostos por Krieg-Brueckner e Luckham [6]. Nesse dois casos, cremos que as asserções ficam ligadas demais ao código, de modo que poderiam constituir um nível adicional, entre os nossos níveis 2 e 3. Com isso as asserções teriam seus próprios comentários; estes serviriam para esclarecer em um nível informal o conteúdo matemático daquelas, o que daria a documentação do desenvolvimento delas e do programa. A documentação desse desenvolvimento não é a intenção exposta em [6]. Além disso, nosso sistema tem a vantagem adicional de documentar as versões.

É interessante observar que as considerações de Beckman [7] contrárias ao uso de comentários em programas não se aplicam às propostas aqui apresentadas, pois os comentários que propomos estão intimamente ligados à funcionalidade dos programas e de seus trechos, ao desenvolvimento dos mesmos e ao uso do nível 1 independentemente dos outros níveis.

Testamos e desenvolvemos estas idéias com alunos de nossa disciplina de Compilação, em que eles elaboram individualmente um

programa de milhares de linhas. Apesar do protesto inicial dos alunos, estes logo perceberam as enormes vantagens do sistema de documentação. Em 1987, apenas 22% da turma concluiu o projeto durante o semestre; em 1988, na turma em que testamos o método, 65% dos alunos o concluiu, provavelmente devido, em boa parte, à documentação.

Várias melhorias podem ser introduzidas no sistema. Por exemplo, pode-se eliminar o aparecimento dos números dos níveis 1 e 2 nas listagens desses níveis, bastando para isso enquadrar totalmente os comentários de nível 1 do programa principal e de cada procedimento em um retângulo de asteriscos. Como vimos, a possibilidade de se documentar as alterações de uma versão para outra pode servir de base para documentar todo processo de desenvovimento do programa por refinamentos sucessivos; além disso os comentários de nível 2 poderiam ser subdivididos em vários subníveis, para produzir uma documentação completa desse desenvolvimento.

O colega Arnaldo Mandel propôs um sistema de 5 níveis. O nível 1 contém a documentação para o usuário "externo", do sistema como um todo, como se ele só tivesse o programa-objeto; o nível 2 contém informações para a compilação; o nível 3 é uma documentação para reutilização das rotinas, com variáveis globais, parâmetros, modo de chamar os procedimentos, etc.; o nível 4 é correspondente ao nosso nível 2; o nível 5 contém o histórico de alterações correspondente aos nossos comentários especiais (ele não propõe que se documente as alterações propriamente ditas).

RobertoC.Mayer implementou o listador em C, possibilitando que a especificação de nível (1 ou 2) valha para uma sequência de linhas, e não somente para cada linha: o encerramento do comentário fecha a sequência. Além disso, sua codificação de comentário de um desses níveis pode ser colocada em qualquer posição de uma linha, ao contrário da nossa proposta, que exige essa colocação no início da linha, simplificando a construção do listador. Seu sistema exige, obviamente, um analisador léxico para detetar os níveis e o encerramento dos comentários correspondentes.

Mirza Neuman propôs o armazenamento apenas da última versão da documentação e dos arquivos com as alterações de uma versão para outra, gerando-se uma versão anterior a partir desses dados. Parece-nos que é possível construir-se esse programa gerador, pois todos os dados necessários estão disponíveis.

8. AGRADECIMENTOS.

O sistema aqui apresentado foi desenvolvido a partir de idéias lançadas por Marcos Ximenes para o sistema ZIM. Especial agradecimento é devido a Afonso C.R. Mastrelli, que programou o exemplo dado no apêndice, com enorme paciência de introduzir paulatinamente as nossas seguidas sugestões. Wagner T. Martins programou uma excelente versão do listador em NATURAL para o sistema ADABAS.

9. REFERÊNCIAS.

[1]Rex, J. - Keeping Track: Five Version Control Systems - Computer Language 5, 6 (June 1988), pp. 111-122.

[2]Setzer, V.W. e Marussi, E. - LDT: Um Gerador Universal de Aplicações para Processamento de Dados - RT-MAC-8806, Departamento de Ciência da Computação do IME-USP (junho de 1988).

[3]Bjorner, D. - Programming Languages: Formal Development of Interpreters and Compilers - in Proc. Intl. Comp. Symp. ICS'77, North Holland Publ., pp. 1-21.

[4]Knuth, D. - Literate Programming - Computer Journal 27, 2 (1984), pp. 97-111.

[5]Gries, D. - The Science of Programming - Springer Verlag, N.York (1981).

[6]Krieg-Brueckner, B. e Luckham, D.C. - Anna: Towards a Language for Annotating Ada Programs - ACM SIGPLAN Notices 15, 11 (Nov. 1980), pp. 128-138.

[7]Beckman, A. - Comments Considered Harmful - ACM SIGPLAN Notices 12, 4 (April 1977), pp. 94-97.

```
 1 1{==========================================================================
 2 1 TITULO: COMENTARIO
 3 1 AUTOR : AFONSO CELSO ROCHA MASTRELLI
 4 1 DATA  : 29/06/88
 5 1 VERSAO: 1.2
 6 1 FINALIDADE : GERAR RELATORIO CONTENDO A DOCUMENTACAO DE DETERMINADO PRO-
 7 1              GRAMA;
 8 1 ENTRADAS : ARQUIVO QUE CONTEM O CODIGO FONTE (EM PASCAL, COBOL OU ZIM)
 9 1            COMENTADO POR NIVEIS;  CODIGO DO NIVEL DE COMENTARIO DESEJADO;
10 1 SAIDAS   : RELATORIO COM A DOCUMENTACAO EXTRAIDA DO ARQUIVO FONTE;
11 1 ROTINAS CHAMADAS: LIB$DO_COMMAND (EXTERNA), MOSTRA_TELA, CABECALHO,
12 1                   RELAT, RELATPAS E SYS$ASCTIM (EXTERNA);
13 1{==========================================================================
51 1{==========================================================================
52 1 ROTINA: MOSTRA_TELA;
53 1 FINALIDADE: MOSTRA MENU AO USUARIO, PERGUNTANDO E RECEBENDO O NIVEL DE
54 1             DOCUMENTACAO QUE ELE QUER E O NOME DO ARQUIVO A SER LIDO,
55 1             PASSANDO ESSAS INFORMACOES AO PROGRAMA PRINCIPAL;
56 1 ENTRADAS: NIVEL DE DOCUMENTACAO E NOME DO ARQUIVO A SER LIDO;
57 1 EXPORTACOES GLOBAIS: NIVEL DE DOCUMENTACAO E NOME DO ARQUIVO A SER LIDO;
58 1 CHAMADO POR: COMENTARIO;
59 1{==========================================================================
111 1{=========================================================================
112 1 ROTINA: CABECALHO;
113 1 FINALIDADE: IMPRIMIR CABECALHO NO INICIO DE CADA PAGINA DO RELATORIO;
114 1 IMPORTACOES GLOBAIS: NOME DO ARQUIVO LIDO, NIVEL DE DOCUMENTACAO,
115 1                      DATA E HORA ATUAIS;
116 1 SAIDA: LINHA DE CABECALHO IMPRESSA NO ALTO DE CADA PAG. DA LISTAGEM;
117 1 CHAMADO POR: COMENTARIO;
118 1{=========================================================================
142 1{=========================================================================
143 1 ROTINA: RELAT;
144 1 FINALIDADE: FAZ LEITURA DO ARQUIVO DE ENTRADA (EM COBOL OU ZIM) E MONTA O
145 1             DE SAIDA;
146 1 IMPORTACOES: CARACTERE QUE IDENTIFICA A LINHA DE COMENTARIO NA LINGUAGEM
147 1              UTILIZADA (COBOL OU ZIM);
148 1 SAIDA  : COMENTARIOS DO PROGRAMA DE ENTRADA NO NIVEL DESEJADO;
149 1 CHAMADO POR: COMENTARIO;
150 1{=========================================================================
206 1{=========================================================================
207 1 ROTINA: RELATPAS;
208 1 FINALIDADE: FAZ LEITURA DO ARQUIVO DE ENTRADA (EM PASCAL) E MONTA O DE
209 1             SAIDA;
210 1 SAIDA: COMENTARIOS DO PROGRAMA DE ENTRADA NO NIVEL DESEJADO;
211 1 CHAMADO POR: COMENTARIO;
212 1{=========================================================================
```

```
  1 1{=============================================================================
  2 1 TITULO: COMENTARIO
  3 1 AUTOR : AFONSO CELSO ROCHA MASTRELLI
  4 1 DATA  : 29/06/88
  5 1 VERSAO: 1.2
  6 1 FINALIDADE : GERAR RELATORIO CONTENDO A DOCUMENTACAO DE DETERMINADO PRO-
  7 1              GRAMA;
  8 1 ENTRADAS : ARQUIVO QUE CONTEM O CODIGO FONTE (EM PASCAL, COBOL OU ZIM)
  9 1            COMENTADO POR NIVEIS;  CODIGO DO NIVEL DE COMENTARIO DESEJADO;
 10 1 SAIDAS   : RELATORIO COM A DOCUMENTACAO EXTRAIDA DO ARQUIVO FONTE;
 11 1 ROTINAS CHAMADAS: LIB$DO_COMMAND (EXTERNA), MOSTRA_TELA, CABECALHO,
 12 1                   RELA , RELATPAS E SYS$ASCTIM (EXTERNA);
 13 1=============================================================================
 51 1{=============================================================================
 52 1 ROTINA: MOSTRA_TELA;
 53 1 FINALIDADE: MOSTRA MENU AO USUARIO, PERGUNTANDO E RECEBENDO O NIVEL DE
 54 1              DOCUMENTACAO QUE ELE QUER E O NOME DO ARQUIVO A SER LIDO,
 55 1              PASSANDO ESSAS INFORMACOES AO PROGRAMA PRINCIPAL;
 56 1 ENTRADAS: NIVEL DE DOCUMENTACAO E NOME DO ARQUIVO A SER LIDO;
 57 1 EXPORTACOES GLOBAIS: NIVEL DE DOCUMENTACAO E NOME DO ARQUIVO A SER LIDO;
 58 1 CHAMADO POR: COMENTARIO;
 59 1=============================================================================
 61 2 INICIO DE MOSTRA_TELA;
 63 2 LIMPA A TELA
 65 2 MOSTRA MENU DE OPCOES AO USUARIO
 84 2 LE OPCAO DO USUARIO
 86 2 OPCAO LIDA E' DE CONTINUAR A EXECUCAO ?
 91 2       LE NOME DO ARQUIVO DE ENTRADA
 95 2       LE NOME DA LINGUAGEM UTILIZADA NO PROGRAMA
 97 2       LINGUAGEM NAO E' COBOL NEM ZIM?
101 2         ASSUME-SE PASCAL COMO LINGUAGEM UTILIZADA
106 2 FIM-DE-MOSTRA-TELA
111 1=============================================================================
112 1 ROTINA: CABECALHO;
113 1 FINALIDADE: IMPRIMIR CABECALHO NO INICIO DE CADA PAGINA DO RELATORIO;
114 1 IMPORTACOES GLOBAIS: NOME DO ARQUIVO LIDO, NIVEL DE DOCUMENTACAO,
115 1                      DATA E HORA ATUAIS;
116 1 SAIDA: LINHA DE CABECALHO IMPRESSA NO ALTO DE CADA PAG. DA LISTAGEM;
117 1 CHAMADO POR: COMENTARIO;
118 1=============================================================================
121 2 INICIO DE CABECALHO;
122 2 PAGINA ESTA' CHEIA ?
125 2       ATUALIZA O NUMERO DA PAGINA.
127 2       RELATORIO JA' FOI INICIALIZADO ?
129 2           SALTA PAGINA
131 2       ESCREVE LINHA DE CABECALHO;
138 2 FIM DE CABECALHO;
142 1=============================================================================
143 1 ROTINA: RELAT;
144 1 FINALIDADE: FAZ LEITURA DO ARQUIVO DE ENTRADA (EM COBOL OU ZIM) E MONTA O
145 1              DE SAIDA;
146 1 IMPORTACOES: CARACTERE QUE IDENTIFICA A LINHA DE COMENTARIO NA LINGUAGEM
147 1               UTILIZADA (COBOL OU ZIM);
```

```
  148 1 SAIDA  : COMENTARIOS DO PROGRAMA DE ENTRADA NO NIVEL DESEJADO;
  149 1 CHAMADO POR: COMENTARIO;
  150 1{=========================================================================}
  155 2 INICIO DE RELAT;
  159 2 PARA CADA LINHA DO ARQUIVO DE ENTRADA :
  162 2     LE LINHA SEQUENCIALMENTE
  164 2     NUMERA A LINHA LIDA
  166 2     DOCUMENTACAO E' DE NIVEL 3 ?
  169 2           ESCREVE LI  IA NC RELATORIO
  174 2       CASO CONTRARIO
  177 2           LINHA NAO E' NULA ?
  180 2                 LINHA E' DE COMENTARIO NIVEL 1 ?
  185 2                       ESCREVE LINHA NO RELATORIO
  188 2                   CASO CONTRARIO
  190 2                     E' DE COMENTARIO NIVEL 2 E USUARIO QUER NIVEL 2 ?
  195 2                         ESCREVE LINHA NO RELATORIO
  202 2 FIM DE RELAT
  206 1{=========================================================================}
  207 1 ROTINA: RELATPAS;
  208 1 FINALIDADE: FAZ LEITURA DO ARQUIVO DE ENTRADA (EM PASCAL) E MONTA O DE
  209 1             SAIDA;
  210 1 SAIDA: COMENTARIOS DO PROGRAMA DE ENTRADA NO NIVEL DESEJADO;
  211 1 CHAMADO POR: COMENTARIO;
  212 1{=========================================================================}
  214 2 INICIO DE RELATPAS
  216 2 PARA CADA LINHA DO ARQUIVO DE ENTRADA :
  219 2     LE LINHA SEQUENCIALMENTE
  222 2     DOCUMENTACAO E' DE NIVEL 3 ?
  225 2           ESCREVE LINHA NO RELATORIO
  230 2       CASO CONTRARIO
  233 2           LINHA NAO E' NULA ?
  236 2                 LINHA E' DE COMENTARIO NIVEL 1 ?
  241 2                       ESCREVE LINHA NO RELATORIO
  244 2                   CASO CONTRARIO
  247 2                       E' DE COMENTARIO NIVEL 2 E USUARIO QUER NIVEL 2 ?
  252 2                           ESCREVE LINHA NO RELATORIO
  260 2 FIM DE RELATPAS
  264 2{-------------------------------------------------------------------------}
  265 2                  P R O G R A M A   P R I N C I P A L
  266 2{-------------------------------------------------------------------------}
  267 2 INICIO DE COMENTARIO
  272 2 EXIBE TELA DE ENTRADA
  274 2 OPCAO DO USUARIO E' CONTINUAR ?
  277 2       ABRE ARQUIVOS DE ENTRADA E SAIDA
  289 2 OPCAO DO USUARIO E' NIVEL 1 OU 2 ?
  294 2             ENTRADA ESTA' EM PASCAL ?
  296 2                   PROCESSA ARQUIVO EM PASCAL
  298 2               CASO CONTRARIO
  300 2                 PROCESSA ARQUIVO EM COBOL OU ZIM
  310 2 ALGUM ARQUIVO FOI ABERTO ?
  313 2       FECHA ARQUIVOS UTILIZADOS
  317 2 FIM DE COMENTARIO
```

```
 1 (*1=============================================================================*)
 2 (*1 TITULO: COMENTARIO                                                          *)
 3 (*1 AUTOR : AFONSO CELSO ROCHA MASTRELLI                                        *)
 4 (*1 DATA  : 29/06/88                                                            *)
 5 (*1 VERSAO: 1.2                                                                 *)
 6 (*1 FINALIDADE : GERAR RELATORIO CONTENDO A DOCUMENTACAO DE DETERMINADO PRO-    *)
 7 (*1               GRAMA;                                                        *)
 8 (*1 ENTRADAS : ARQUIVO QUE CONTEM O CODIGO FONTE (EM PASCAL, COBOL OU ZIM)      *)
 9 (*1            COMENTADO POR NIVEIS;  CODIGO DO NIVEL DE COMENTARIO DESEJADO;  *)
10 (*1 SAIDAS   : RELATORIO COM A DOCUMENTACAO EXTRAIDA DO ARQUIVO FONTE;          *)
11 (*1 ROTINAS CHAMADAS: LIB$DO_COMMAND (EXTERNA), MOSTRA_TELA, CABECALHO,        *)
12 (*1                   RELAT, RELATPAS E SYS$ASCTIM (EXTERNA);                  *)
13 (*1=============================================================================*)
14 PROGRAM COMENTARIO(INPUT,OUTPUT);
15
16 CONST
17     CLS = CHR(27) + '[H' + CHR(27) + '[J'; (* LIMPA A TELA DO VT100/200/220 *)
18 TYPE
19     LINHA      = VARYING[120] OF CHAR; (* PARA LINHA DE IMPRESSAO;          *)
20     CARACTERE  = CHAR;                 (* PARA PARAMETRO DE RELAT;          *)
21     $WORD      = [WORD] 0..65535;      (* PARA PARAMETRO DE SYS$ASCTIM;     *)
22     $DATE_TIME = [QUAD] RECORD END;    (* PARA PARAMETRO DE SYS$ASCTIM;     *)
23
24 VAR
25     COMANDO : VARYING[200] OF CHAR; (* COMANDO DCL A SER EXECUTADO;         *)
26     LINPROG : LINHA;                (* GUARDA LINHA DO PROGRAMA LIDO;       *)
27     OPCAO   : INTEGER;              (* NRO DA OPCAO DO USUARIO NO MENU;     *)
28     NOMEARQ : VARYING[30] OF CHAR;  (* NOME DO ARQUIVO QUE CONTEM O PROGRAMA; *)
29     SAIDA   : TEXT;                 (* CONTEM O RELATORIO DE SAIDA;         *)
30     ENTRADA : TEXT;                 (* PROGRAMA COMENTADO A SER LIDO;       *)
31     LING    : CARACTERE;            (* LINGUAGEM DO PROGRAMA A SER LIDO;    *)
32     NUMLIN  : INTEGER;              (* NUMERACAO DAS LINHAS DO FONTE LIDO;  *)
33     LINPAG  : INTEGER;              (* NUMERO DE LINHAS POR PAGINA;         *)
34     NPAG    : INTEGER;              (* NUMERO DA PAGINA DO RELATORIO;       *)
35     DATA_HORA: VARYING[17] OF CHAR; (* DATA E HORA RECEBIDA DO SISTEMA;     *)
36     RETORNO : INTEGER;              (* SOMENTE AUXILIA O RECEBIMENTO DA HORA; *)
37
38
39 (* ROTINA EXTERNA QUE EXECUTA COMANDO DCL                                    *)
40 PROCEDURE LIB$DO_COMMAND(CMDTXT : VARYING[A] OF CHAR); EXTERN;
41
42
43 (* FUNCAO EXTERNA QUE RETORNA DATA E HORA ATUAIS DO SISTEMA OPERACIONAL      *)
44 [EXTERNAL] FUNCTION SYS$ASCTIM(%REF NADA1 : $WORD := %IMMED 0;
45                                DATA_HORA : [CLASS_S]
46                                 PACKED ARRAY[1..U:INTEGER] OF CHAR;
47                                %REF NADA2 : $DATE_TIME := %IMMED 0;
48                                NADA3 : UNSIGNED := %IMMED 0):INTEGER; EXTERN;
49
50
51 (*1=============================================================================*)
52 (*1 ROTINA: MOSTRA_TELA;                                                        *)
53 (*1 FINALIDADE: MOSTRA MENU AO USUARIO, PERGUNTANDO E RECEBENDO O NIVEL DE     *)
```

```
   54 (*1              DOCUMENTACAO QUE ELE QUER E O NOME DO ARQUIVO A SER LIDO,     *)
   55 (*1              PASSANDO ESSAS INFORMACOES AO PROGRAMA PRINCIPAL;             *)
   56 (*1 ENTRADAS: NIVEL DE DOCUMENTACAO E NOME DO ARQUIVO A SER LIDO;              *)
   57 (*1 EXPORTACOES GLOBAIS: NIVEL DE DOCUMENTACAO E NOME DO ARQUIVO A SER LIDO;   *)
   58 (*1 CHAMADO POR: COMENTARIO;                                                   *)
   59 (*1=============================================================================*)
   60 PROCEDURE MOSTRA_TELA;
   61 (*2 INICIO DE MOSTRA_TELA;                                                    *)
   62 BEGIN
   63 (*2 LIMPA A TELA                                                              *)
   64     WRITE(CLS);
   65 (*2 MOSTRA MENU DE OPCOES AO USUARIO                                          *)
   66     WRITELN(' ':25,'DOCUMENTACAO DE SOFTWARE');
   67     WRITELN(' ':25,'^^^^^^^^^^^^^^^^^^^^^^^^');
   68     WRITELN;
   69     WRITELN(' ':10,'VOCE PODE LISTAR 3 NIVEIS DE DOCUMENTACAO DE UM PROGRAMA.');
   70     WRITELN(' ':10,'O PROGRAMA PODE ESTAR NUMA DAS SEGUINTES LINGUAGENS: ');
   71     WRITELN;
   72     WRITELN(' ':27,'[P]ASCAL   (DEFAULT)');
   73     WRITELN(' ':27,'[C]OBOL');
   74     WRITELN(' ':27,'[Z]IM');
   75     WRITELN; WRITELN;
   76     WRITELN(' ':26,'NIVEIS DE DOCUMENTACAO');
   77     WRITELN;
   78     WRITELN(' ':12,'1. DESCRICAO FUNCIONAL + FINALIDADES');
   79     WRITELN(' ':12,'2. NIVEL 1 + PSEUDO-CODIGO');
   80     WRITELN(' ':12,'3. NIVEIS 1 E 2 + CODIGO FONTE');
   81     WRITELN(' ':12,'4. F I M');
   82     WRITELN; WRITELN;
   83     WRITE(' ':15,'SUA OPCAO E''          : ');
   84 (*2 LE OPCAO DO USUARIO                                                      *)
   85     READLN(OPCAO);
   86 (*2 OPCAO LIDA E' DE CONTINUAR A EXECUCAO ?                                  *)
   87     IF ((OPCAO = 1) OR (OPCAO = 2) OR (OPCAO = 3 )) THEN
   88       BEGIN
   89           WRITELN;
   90           WRITE(' ':15,'NOME DO ARQUIVO     : ');
   91 (*2        LE NOME DO ARQUIVO DE ENTRADA                                     *)
   92           READLN(NOMEARQ);
   93           WRITELN;
   94           WRITE(' ':15,'LINGUAGEM UTILIZADA : ');
   95 (*2        LE NOME DA LINGUAGEM UTILIZADA NO PROGRAMA                        *)
   96           READLN(LING);
   97 (*2        LINGUAGEM NAO E' COBOL NEM ZIM?                                   *)
   98           CASE LING OF
   99             'C','C' : LING := 'C';
  100             'Z','Z' : LING := 'Z';
  101 (*2          ASSUME-SE PASCAL COMO LINGUAGEM UTILIZADA                       *)
  102             OTHERWISE
  103                LING := 'P';
  104             END; (* CASE *)
  105         END; (* IF *)
  106 (*2 FIM-DE-MOSTRA-TELA                                                       *)
  107 END; (* MOSTRA TELA *)
```

```
108
109
110
111 (*1=============================================================*)
112 (*1 ROTINA: CABECALHO;                                           *)
113 (*1 FINALIDADE: IMPRIMIR CABECALHO NO INICIO DE CADA PAGINA DO RELATORIO;  *)
114 (*1 IMPORTACOES GLOBAIS: NOME DO ARQUIVO LIDO, NIVEL DE DOCUMENTACAO,      *)
115 (*1                     DATA E HORA ATUAIS;                     *)
116 (*1 SAIDA: LINHA DE CABECALHO IMPRESSA NO ALTO DE CADA PAG. DA LISTAGEM;   *)
117 (*1 CHAMADO POR: COMENTARIO;                                     *)
118 (*1=============================================================*)
119 PROCEDURE CABECALHO;
120 BEGIN
121 (*2 INICIO DE CABECALHO;                                         *)
122 (*2 PAGINA ESTA' CHEIA ?                                         *)
123     IF LINPAG > 57 THEN
124         BEGIN
125 (*2       ATUALIZA O NUMERO DA PAGINA.                           *)
126           NPAG := NPAG + 1;
127 (*2       RELATORIO JA' FOI INICIALIZADO ?                       *)
128           IF NPAG > 1 THEN
129 (*2           SALTA PAGINA                                       *)
130               PAGE(SAIDA);
131 (*2       ESCREVE LINHA DE CABECALHO;                            *)
132           WRITELN(SAIDA); WRITELN(SAIDA);
133           WRITE(SAIDA,'    PROGRAMA: ',NOMEARQ,' ':(22-NOMEARQ.LENGTH));
134           WRITE(SAIDA,'DOC.NIVEL ',OPCAO:2,'    DATA: ',DATA_HORA,'      PAG: ',NPAG:4);
135           WRITELN(SAIDA); WRITELN(SAIDA);
136           LINPAG := 4;
137           END; (* IF *)
138 (*2 FIM DE CABECALHO;                                           *)
139 END; (* CABECALHO *)
140
141
142 (*1=============================================================*)
143 (*1 ROTINA: RELAT;                                              *)
144 (*1 FINALIDADE: FAZ LEITURA DO ARQUIVO DE ENTRADA (EM COBOL OU ZIM) E MONTA O *)
145 (*1             DE SAIDA;                                       *)
146 (*1 IMPORTACOES: CARACTERE QUE IDENTIFICA A LINHA DE COMENTARIO NA LINGUAGEM  *)
147 (*1               UTILIZADA (COBOL OU ZIM);                     *)
148 (*1 SAIDA  : COMENTARIOS DO PROGRAMA DE ENTRADA NO NIVEL DESEJADO;  *)
149 (*1 CHAMADO POR: COMENTARIO;                                     *)
150 (*1=============================================================*)
151 PROCEDURE RELAT(INI : CARACTERE);
152 VAR
153    IDENT1, IDENT2 : VARYING[2] OF CHAR;      (* CONTEM IDENTIFICADOR DE NIVEL *)
154
155 (*2 INICIO DE RELAT;                                            *)
156 BEGIN
157     IDENT1 := INI + '1';
158     IDENT2 := INI + '2';
159 (*2 PARA CADA LINHA DO ARQUIVO DE ENTRADA :                      *)
160     WHILE NOT EOF(ENTRADA) DO
161       BEGIN
```

```
  (*2     LE LINHA SEQUENCIALMENTE                                          *)
          READLN(ENTRADA,LINPROG);
  (*2     NUMERA A LINHA LIDA                                               *)
          NUMLIN := NUMLIN + 1;
  (*2     DOCUMENTACAO E' DE NIVEL 3 ?                                      *)
          IF OPCAO = 3 THEN
              BEGIN
  (*2           ESCREVE LINHA NO RELATORIO                                  *)
                LINPAG := LINPAG + 1;
                CABECALHO;
                WRITELN(SAIDA,' ':4,NUMLIN:5,' ',LINPROG);
                END
  (*2         CASO CONTRARIO                                                *)
              ELSE
              BEGIN
  (*2           LINHA NAO E' NULA ?                                         *)
                IF (LENGTH(LINPROG) > 2) THEN
                    BEGIN
  (*2               LINHA E' DE COMENTARIO NIVEL 1 ?                        *)
                    IF (SUBSTR(LINPROG,1,2) = IDENT1) THEN
                        BEGIN
                          LINPAG := LINPAG + 1;
                          CABECALHO;
  (*2                     ESCREVE LINHA NO RELATORIO                        *)
                          WRITELN(SAIDA,' ':4,NUMLIN:5,' ',SUBSTR(LINPROG,2,LENGTH(LINPROG)-1));
                        END
                      CASO CONTRARIO                                        *)
                      ELSE
  (*2                   E' DE COMENTARIO NIVEL 2 E USUARIO QUER NIVEL 2 ?   *)
                      IF ((SUBSTR(LINPROG,1,2) = IDENT2) AND (OPCAO = 2)) THEN
                          BEGIN
                            LINPAG := LINPAG + 1;
                            CABECALHO;
  (*2                       ESCREVE LINHA NO RELATORIO                      *)
                            WRITELN(SAIDA,' ':4,NUMLIN:5,' ',SUBSTR(LINPROG,2,LENGTH(LINPROG)-1));
                            END;
                    END; (* IF *)
                END; (* IF *)
          END; (* WHILE *)
          WRITELN(SAIDA);
  (*2 FIM DE RELAT                                                          *)
END; (* RELAT *)


(*1=============================================================================*)
(*1 ROTINA: RELATPAS;                                                          *)
(*1 FINALIDADE: FAZ LEITURA DO ARQUIVO DE ENTRADA (EM PASCAL) E MONTA O DE     *)
(*1              SAIDA;                                                       *)
(*1 SAIDA: COMENTARIOS DO PROGRAMA DE ENTRADA NO NIVEL DESEJADO;               *)
(*1 CHAMADO POR: COMENTARIO;                                                   *)
(*1=============================================================================*)
PROCEDURE RELATPAS;
(*2 INICIO DE RELATPAS                                                       *)
BEGIN
```

```
(*2 PARA CADA LINHA DO ARQUIVO DE ENTRADA :                                  *)
    WHILE NOT EOF(ENTRADA) DO
      BEGIN
(*2     LE LINHA SEQUENCIALMENTE                                             *)
        READLN(ENTRADA,LINPROG);
        NUMLIN := NUMLIN + 1;
(*2     DOCUMENTACAO E' DE NIVEL 3 ?                                         *)
        IF OPCAO = 3 THEN
            BEGIN
(*2           ESCREVE LINHA NO RELATORIO                                     *)
              LINPAG := LINPAG + 1;
              CABECALHO;
              WRITELN(SAIDA,' ':4,NUMLIN:5,' ',LINPROG);
            END
(*2       CASO CONTRARIO                                                     *)
        ELSE
          BEGIN
(*2         LINHA NAO E' NULA ?                                              *)
            IF (LENGTH(LINPROG) > 3) THEN
                BEGIN
(*2               LINHA E' DE COMENTARIO NIVEL 1 ?                           *)
                  IF (SUBSTR(LINPROG,1,3) = '(*1') THEN
                      BEGIN
                        LINPAG := LINPAG + 1;
                        CABECALHO;
(*2                     ESCREVE LINHA NO RELATORIO                           *)
                        WRITELN(SAIDA,' ':4,NUMLIN:5,' ',SUBSTR(LINPROG,3,INDEX(LINPROG,'*)')-3))
                        END
                    CASO CONTRARIO                                           *)
                    ELSE
                      BEGIN
(*2                     E' DE COMENTARIO NIVEL 2 E USUARIO QUER NIVEL 2 ?    *)
                        IF ((SUBSTR(LINPROG,1,3) = '(*2') AND (OPCAO = 2)) THEN
                            BEGIN
                              LINPAG := LINPAG + 1;
                              CABECALHO;
(*2                           ESCREVE LINHA NO RELATORIO                     *)
                              WRITELN(SAIDA,' ':4,NUMLIN:5,' ',SUBSTR(LINPROG,3,INDEX(LINPROG,'*)')-3));
                              END;
                        END;
                    END; (* IF *)
                END; (* IF *)
          END; (* WHILE *)
    WRITELN(SAIDA);
(*2 FIM DE RELATPAS                                                          *)
END; (* RELATPAS *)


(*2-------------------------------------------------------------------------*)
(*2                    P R O G R A M A   P R I N C I P A L                  *)
(*2-------------------------------------------------------------------------*)
(*2 INICIO DE COMENTARIO                                                     *)
BEGIN
    OPCAO := 0;
```

```
  270      DATA_HORA := '                    ';
  271      NUMLIN := 0;
  272  (*2 EXIBE TELA DE ENTRADA                                              *)
  273      MOSTRA_TELA;
  274  (*2 OPCAO DO USUARIO E' CONTINUAR ?                                    *)
  275      IF ((OPCAO > 0) AND (OPCAO < 4)) THEN
  276          BEGIN
  277  (*2       ABRE ARQUIVOS DE ENTRADA E SAIDA                            *)
  278            OPEN(ENTRADA,NOMEARQ,
  279                 HISTORY := OLD,
  280                 ACCESS_METHOD := SEQUENTIAL);
  281                 OPEN(SAIDA,'COMMENT.LIS',HISTORY := NEW);
  282           RESET(ENTRADA);
  283           REWRITE(SAIDA);
  284           RETCRNO := SYS$ASCTIM(,DATA_HORA,,);
  285           (* FORCA A IMPRESSAO DO CABECALHO NA PRIMEIRA PAGINA *)
  286           LINPAG := 70;
  287           CABECALHO;
  288           END; (* IF *)
  289  (*2 OPCAO DO USUARIO E' NIVEL 1 OU 2 ?                                 *)
  290      CASE OPCAO OF
  291        1,2,3 : BEGIN
  292                IF (NOM. .RQ.LENGTH > 20) THEN
  293                    NOMEARQ := SUBSTR(NOMEARQ,1,20);
  294  (*2           ENTRADA ESTA' EM PASCAL ?                               *)
  295                IF (LING = 'P') THEN
  296  (*2             PROCESSA ARQUIVO EM PASCAL                                *)
  297                  RELATPAS
  298  (*2           CASO CONTRARIO                                          *)
  299                ELSE
  300  (*2             PROCESSA ARQUIVO EM COBOL OU ZIM                      *)
  301                  IF (LING = 'C') THEN
  302                      RELAT('*')
  303                  ELSE
  304                      RELAT('%');
  305  (*           MANDA RELATORIO PARA A FILA DE IMPRESSAO                  *)
  306  (*             COMANDO := 'PRINT COMMENT.LIS';  *)
  307  (*             LIB$DO_COMMAND(COMANDO);         *)
  308                END; (* 1,2,3 *)
  309        END; (* CASE *)
  310  (*2 ALGUM ARQUIVO FOI ABERTO ?                                          *)
  311      IF ((OPCAO > 0) AND (OPCAO < 4)) THEN
  312          BEGIN
  313  (*2       FECHA ARQUIVOS UTILIZADOS                                   *)
  314            CLOSE(ENTRADA);
  315            CLOSE(SAIDA);
  316            END;
  317  (*2 FIM DE COMENTARIO                                                  *)
  318 END. (* COMENTARIO *)
```