

# Turbo Pascal

# Turbo Pascal

- Contém todas as informações necessárias para utilizar o Turbo Pascal versão 5.0, numa linguagem fácil de entender.
- Perfeito para carregar de um lado para outro.
- É a ferramenta ideal para quem utiliza o Turbo Pascal.

ISBN 85-85184-41-8



9 788585 184414

## GUIA DE CONSULTA RÁPIDA

Dennis Cintra Leite  
Rubens Prates

Direitos reservados pela Novatec Editora Ltda.  
Rua Cons. Moreirã de Barros 1084 Conj. 01/02  
Caixa Postal 12191-6 CEP 02098-970 São Paulo Brasil  
Fone (011) 959-6529 Fax (011) 950-8869  
Internet: <http://www.novatec1.com>

www.novatec1.com  
**novatec** EDITORA

NOVOS PREFIXOS  
TELEFONE (011) 6959-6529  
FAX (011) 6950-8869

Complete a sua coleção!

Outubro/98

ARJ  
C++  
CLIPPER versão 5.2  
CLIPPER Mensagens de Erro versões 5.01/5.2  
COBOL ANS 85  
COBOL Sem Mistérios  
Construindo Aplicações com o Visual Basic 4  
CORELDRAW 5.0 (software em inglês)  
CORELDRAW FONTS versão 7.0  
CORELDRAW SYMBOLS versão 7.0  
eBASE III Plus  
EMOTICONS  
EUDORA LIGHT  
EXCEL versão 4.0 (software em português)  
HTML

## Menu Principal do Turbo Pascal

|                                |                                                                                   |
|--------------------------------|-----------------------------------------------------------------------------------|
| <b>File</b>                    | Manipulação de arquivos.                                                          |
| <b>Load (F3)</b>               | Carrega um arquivo.                                                               |
| <b>Pick (ALT+F3)</b>           | Permite selecionar um dos oito últimos arquivos editados.                         |
| <b>New</b>                     | Apaga o programa corrente.                                                        |
| <b>Save (F2)</b>               | Grava o programa corrente com o nome mostrado na linha de status.                 |
| <b>Write to</b>                | Grava o programa corrente. Pede o nome do arquivo.                                |
| <b>Directory</b>               | Mostra o diretório corrente.                                                      |
| <b>Change dir</b>              | Altera o diretório corrente.                                                      |
| <b>OS shell</b>                | Transfere o controle para o DOS. Para retornar digite EXIT.                       |
| <b>Quit (ALT+X)</b>            | Sai do Turbo Pascal e retorna ao DOS.                                             |
| <b>Edit</b>                    | Entra no editor do Turbo Pascal.                                                  |
| <b>Run</b>                     | Executa o programa corrente (compila e linkedita, se necessário).                 |
| <b>Run (CTRL+F9)</b>           | Executa o programa.                                                               |
| <b>Program reset (CTRL+F2)</b> | Reinicia o depurador, desaloca memória, fecha arquivos, conservando as variáveis. |
| <b>Go to cursor (F4)</b>       | Executa até a posição do cursor.                                                  |
| <b>Trace into (F7)</b>         | Acompanha a execução dentro de sub-rotinas ou arquivos "include".                 |
| <b>Step over (F8)</b>          | Não executa sub-rotinas ou includes.                                              |
| <b>User screen (ALT+F5)</b>    | Mostra a tela de saída do programa.                                               |
| <b>Compile</b>                 | Opções de compilação.                                                             |
| <b>Compile (ALT+F9)</b>        | Compila o programa ativo.                                                         |
| <b>Make (F9)</b>               | Recompila o programa, caso qualquer subunidade tenha sido alterada.               |
| <b>Build</b>                   | Recompila o programa e suas subunidades incondicionalmente.                       |
| <b>Destination</b>             | Destino do código executável (memória ou disco).                                  |
| <b>Find error</b>              | Localiza erro no programa fonte, dado o endereço (segmento:deslocamento).         |
| <b>Primary file</b>            | Define o arquivo principal para compilação.                                       |
| <b>Get info</b>                | Mostra informações sobre o arquivo ativo.                                         |
| <b>Options</b>                 | Opções do ambiente integrado.                                                     |
| <b>Compiler</b>                | Opções de compilação.                                                             |
| <b>Range checking</b>          | Verifica faixa de subscritos; {SR}.                                               |
| <b>Stack checking</b>          | Verifica espaço na pilha; {SS}.                                                   |
| <b>I/O checking</b>            | Verifica os erros de E/S; {SI}.                                                   |
| <b>Force far calls</b>         | Força chamadas FAR; {SF}.                                                         |
| <b>Overlays allowed</b>        | Gera código que permite overlays; {SO+}.                                          |
| <b>Align Data</b>              | Alinha variáveis em endereços pares; {SA}.                                        |
| <b>Var-string checking</b>     | Relaxa a verificação de parâmetros string; {SV}.                                  |
| <b>Boolean evaluation</b>      | Avaliação completa de expressões booleanas; {SB}.                                 |
| <b>Numeric processing</b>      | Gera código para o 80x87; {SN}.                                                   |
| <b>Emulation</b>               | Gera emulação em software do 80x87; {SN}.                                         |
| <b>Debug information</b>       | Gera informações para debug; {SD}.                                                |
| <b>Local symbols</b>           | Gera símbolos locais para debug; {SL}.                                            |
| <b>Conditional defines</b>     | Define símbolos para compilação condicional (separados por ";").                  |
| <b>Memory sizes</b>            | Configura a alocação de memória.                                                  |
| <b>Stack size</b>              | Tamanho da pilha (0-64K, default=16K).                                            |
| <b>Low heap limit</b>          | Tamanho mínimo do heap (default=0K).                                              |
| <b>High heap limit</b>         | Tamanho máximo do heap.                                                           |

Visite o web site da Novatec Editora em <http://www.novatec1.com> e conheça os últimos lançamentos

|                                 |                                                                                              |
|---------------------------------|----------------------------------------------------------------------------------------------|
| <b>Linker</b>                   | Opções do linkeditor.                                                                        |
| <b>Map file</b>                 | Informações no .MAP (se ele tiver sido criado).                                              |
| <b>Segments</b>                 | Somente segmentos (começo/fim, nome, tamanho, classe).                                       |
| <b>Publics</b>                  | Segmentos, símbolos, endereços e ponto de entrada.                                           |
| <b>Detailed Link buffer</b>     | Todas as informações possíveis.                                                              |
| <b>Environment</b>              | Buffer de encadeamento em memória ou disco.                                                  |
| <b>Config auto save</b>         | Localização dos arquivos necessários para compilar, linkeditor e fornecer Help.              |
| <b>Edit auto save</b>           | Gravação automática da configuração ativa quando o DOS é acessado ou o programa é executado. |
| <b>Backup files</b>             | Gravação automática do programa ativo quando o DOS é acessado ou o programa é executado.     |
| <b>Tab size</b>                 | Mantém versão antiga quando grava a atual.                                                   |
| <b>Zoom windows</b>             | Número de espaços por tabulação.                                                             |
| <b>Screen size</b>              | Aumenta/diminui o tamanho da janela.                                                         |
| <b>Directories</b>              | Tamanho da tela (25, 43 ou 50 linhas).                                                       |
| <b>Turbo directory</b>          | Diretórios onde se localizam os arquivos (separados por ";").                                |
| <b>EXE &amp; TPU directory</b>  | Diretório dos arquivos TURBO.TP e TURBO.HLP.                                                 |
| <b>Include directories</b>      | Diretório onde serão criados arquivos .EXE, .TPU e .MAP.                                     |
| <b>Unit directories</b>         | Diretório dos arquivos incluídos com {\$I arq}.                                              |
| <b>Object directories</b>       | Diretório dos arquivos .OBJ incluídos com {\$L arq}.                                         |
| <b>Pick file name</b>           | Arquivo 'Pick' (oito últimas edições).                                                       |
| <b>Current pick file</b>        | Mostra o nome do arquivo 'Pick' ativo.                                                       |
| <b>Parameters</b>               | Simula a linha de comando.                                                                   |
| <b>Save options</b>             | Grava a configuração ativa.                                                                  |
| <b>Retrieve options</b>         | Carrega um arquivo de configuração.                                                          |
| <b>Debug</b>                    | Opções de depuração.                                                                         |
| <b>Evaluate (CTRL+F4)</b>       | Avalia expressões e modifica variáveis.                                                      |
| <b>Call stack (CTRL+F3)</b>     | Pilha de chamada de funções.                                                                 |
| <b>Find procedure</b>           | Acha função ou procedure do programa.                                                        |
| <b>Integrated debugging</b>     | Habilita a depuração. Para ganhar espaço, desabilite esta opção.                             |
| <b>Standalone debugging</b>     | Opções de visualização da tela de saída do programa durante a depuração.                     |
| <b>Display swapping</b>         | Inclui informações de depuração no .EXE.                                                     |
| <b>None</b>                     | Nunca mostra.                                                                                |
| <b>Smart</b>                    | Decido pelo ambiente integrado.                                                              |
| <b>Always</b>                   | Sempre mostra depois de executar um comando.                                                 |
| <b>Refresh display</b>          | Recupera a tela, se sobrescrita.                                                             |
| <b>Break/Watch</b>              | Breakpoints e monitoramento.                                                                 |
| <b>Add watch (CTRL+F7)</b>      | Exibe o valor de uma variável/expressão.                                                     |
| <b>Delete watch</b>             | Retira uma variável ou expressão da janela "watch".                                          |
| <b>Edit watch</b>               | Edita uma variável/expressão em janela.                                                      |
| <b>Remove all watches</b>       | Retira as variáveis/expressões da janela.                                                    |
| <b>Toggle breakp. (CTRL+F8)</b> | Define/elimina um ponto de interrupção no programa.                                          |
| <b>Clear all breakpoints</b>    | Reinicializa os pontos de interrupção do programa.                                           |
| <b>View next breakpoint</b>     | Visualiza o próximo ponto de interrupção.                                                    |

## Diretivas de Compilação

|                          |                                                                                                                                               |
|--------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------|
| <b>{\$A+ -}</b>          | Alinha (+) ou não (-) as variáveis/constantes tipadas em endereços pares. Inócuo no 8088; apressa a execução no 8086, 80286, etc.             |
| <b>{\$B+}</b>            | Gera código para avaliação completa de operações booleanas.                                                                                   |
| <b>{\$B-}</b>            | Não gera código completo para avaliação de operações booleanas. A avaliação termina quando o resultado se torna evidente.                     |
| <b>{\$D+ -}</b>          | Gera (+) ou não (-) informações para o depurador.                                                                                             |
| <b>{\$E+ -}</b>          | Gera (+) ou não (-) código emulador do co-processador 80x87.                                                                                  |
| <b>{\$F+}</b>            | Força a utilização do modelo FAR de memória.                                                                                                  |
| <b>{\$F-}</b>            | O modelo é selecionado automaticamente.                                                                                                       |
| <b>{\$I+}</b>            | Interrompe o programa quando detecta erros de entrada/saída.                                                                                  |
| <b>{\$I-}</b>            | Os erros de entrada/saída são assinalados em IOResult. O programa deve conter suas próprias rotinas de manuseio de erros.                     |
| <b>{\$I arq}</b>         | Inclui arq durante a compilação (até oito níveis de inclusão).                                                                                |
| <b>{\$L+ -}</b>          | Gera (+) ou não (-) código de depuração para símbolos locais à unit.                                                                          |
| <b>{\$L arq}</b>         | Incorpora arq (em linguagem de máquina) ao código gerado.                                                                                     |
| <b>{\$M t,m,n}</b>       | Configura o uso da memória.<br>t (1024 - 65520, default = 16384).<br>m (0 - 655360, default = 0).<br>n (min_heap - 655360, default = 655360). |
| <b>{\$N+}</b>            | Gera código para o co-processador 80x87 no cálculo com números reais; se {\$E+} ativo, dispensa a presença do 80x87.                          |
| <b>{\$N-}</b>            | Gera código compactado para cálculo com números reais.                                                                                        |
| <b>{\$O+ -}</b>          | Gera (+) ou não (-) código que permite overlay.                                                                                               |
| <b>{\$O&lt;unid&gt;}</b> | Transforma uma unidade em overlay.                                                                                                            |
| <b>{\$R+ -}</b>          | Verifica (+) ou não (-) o estouro de índices em matrizes e strings.                                                                           |
| <b>{\$S+ -}</b>          | Verifica (+) ou não (-) a existência de espaço suficiente na pilha para a execução de procedure ou função.                                    |
| <b>{\$V+ -}</b>          | Exige (+) ou não (-) correspondência exata entre os parâmetros formais e reais na passagem de strings às funções ou procedures.               |

## Diretivas de Compilação Condicional

|                              |                                                        |
|------------------------------|--------------------------------------------------------|
| <b>{\$DEFINE símbolo}</b>    | Define o símbolo usado na compilação condicional.      |
| <b>Símbolos Predefinidos</b> |                                                        |
| <b>VER40</b>                 | Versão 4.0.                                            |
| <b>VER50</b>                 | Versão 5.0.                                            |
| <b>VER55</b>                 | Versão 5.5.                                            |
| <b>MSDOS</b>                 | Sistema MS-DOS/PC-DOS.                                 |
| <b>CPU86</b>                 | CPU é da família 80x86.                                |
| <b>CPU87</b>                 | Definida se o 80x87 estiver presente.                  |
| <b>{\$UNDEF símbolo}</b>     | Cancela a definição do símbolo.                        |
| <b>{\$IFDEF símbolo}</b>     | Continua a compilação se o símbolo estiver definido.   |
| <b>{\$IFNDEF símbolo}</b>    | Continua a compilação se símbolo não estiver definido. |
| <b>{\$IFOPT diret+ -}</b>    | Compila o código que segue se a diretiva for + ou -.   |
| <b>{\$ELSE}</b>              | Compila o próximo bloco se o IFxxx anterior falhar.    |
| <b>{\$ENDIF}</b>             | Encerra o bloco de compilação condicional.             |

## Comandos do Turbo Pascal

**procedimento** [(parâmetro,...)];

Executa um procedimento predefinido ou definido no programa.

**função** := expressão;

Dentro de uma definição de função, atribui o valor da expressão à função, de forma que ela retorne esse mesmo valor.

**variável** := expressão;

Atribui o valor da expressão à variável.

**rótulo**:

Antes da linha de comando, identifica o destino de um GOTO.

**BEGIN comando;... END**

Define um bloco de comandos. Se a sintaxe do Turbo Pascal permitir um comando simples, também permitirá um bloco de comandos.

**CASE expressão OF**

opção1 : comando1;

opção2 : comando2;...

[ELSE comando;]

**END;**

Comando condicional múltiplo. Executa só o comando associado à opção que for igual à expressão.

**FOR var := início TO fim DO comando;**

Repetição com var crescendo desde início até fim (todas tipo Ordinal).

**FOR var := fim DOWNTO início DO comando;**

Repetição com var decrescendo desde início até fim (todas tipo Ordinal).

**GOTO rótulo;**

Desvia a execução do programa para o rótulo especificado.

**IF condição THEN comando1 [ELSE comando2];**

Comando condicional. Se a condição for verdadeira o comando1 será executado, caso contrário, o comando2 será executado. No caso de ser usada a opção ELSE, o comando1 não pode ser seguido de \*;.

**INLINE (código/...);**

Insere instruções em linguagem de máquina diretamente no programa fonte.

**REPEAT comando;... UNTIL condição;**

Executa o comando (pelo menos uma vez) até a condição ser verdadeira (True).

**WHILE condição DO comando;**

Executa o comando enquanto a condição for verdadeira.

**WITH variáveis\_record DO comando;**

Permite referências abreviadas a campos de uma variável tipo Record.

## Procedures e Funções do Turbo Pascal

**Abs(x : <número>) : <número> [SYS] (F)**

Retorna o valor absoluto de x.

**Addr(VAR x : <ident>) : <pointer> [SYS] (F)**

Retorna o endereço de x. O mesmo que o operador @.

**Append(VAR f : Text) [SYS] (P)**

Abre um arquivo já existente para inclusão no seu final.

**Arc(x,y : Integer; angi, angf, raio : Word) [GRP] (P)**

Desenha um arco.

x,y Coordenadas do centro do arco.

angi,angf Ângulos inicial e final do arco medido a partir do eixo das abscissas no sentido anti-horário.

**ArcTan(x : <real>) : <real> [SYS] (F)**

Retorna o arco-tangente (em radianos) do ângulo x.

**Assign(VAR f : <file>; nome : String) [SYS] (P)**

Atribui o nome de um arquivo externo a uma variável tipo arquivo.

**AssignCrt(VAR f : Text) [CRT] (P)**

Associa um arquivo-texto f ao vídeo (CRT). Usado para listagens rápidas pois não passa pelo BIOS.

**Bar(x1, y1, x2, y2 : Integer) [GRP] (P)**

Desenha um retângulo com a cor definida em SetFillStyle e padrão definido em SetFillPattern.

**Bar3D(x1,y1,x2,y2:Integer; prof : Word; topo : Boolean) [GRP] (P)**

Desenha um retângulo em três dimensões usando estilo e cor ativos (definidos em SetFillStyle e SetFillPattern).

prof Valor que aumenta a ilusão de profundidade.

topo Se True desenha o topo da barra, se False permite empilhar barras. As seguintes constantes predefinidas estão disponíveis:  
TopOn=True;TopOff=False;

**BlockRead(VAR f : File; VAR buffer; n : Word [;lidos: Word]) [SYS] (P)**

Lê n ou menos registros de um arquivo f armazenando-os a partir do endereço inicial de buffer; opcionalmente retorna o número de registros efetivamente lidos.

**BlockWrite(VAR f : File; VAR buffer; n : Word [;gravados: Word]) [SYS] (P)**

Grava n ou menos registros armazenados a partir do endereço inicial de buffer, no arquivo f; opcionalmente retorna o número de registros efetivamente gravados.

**ChDir(path : String) [SYS] (P)**

Muda o diretório ativo para path.

**Chr(n : Byte) : Char [SYS] (F)**

Retorna o caractere correspondente ao código ASCII n.

**Circle(x, y : Integer; raio : Word) [GRP] (P)**

Desenha um círculo centrado em x,y com diâmetro = 2\*raio.

**ClearDevice [GRP] (P)**

Limpa a tela gráfica.

**ClearViewport [GRP] (P)**

Limpa a viewport (janela gráfica) ativa.

**Close(VAR f : <file>) [SYS] (P)**

Fecha um arquivo aberto f.

**CloseGraph [GRP] (P)**

Volta ao modo de tela anterior à chamada da **InitGraph**, liberando todos os buffers e pilhas alocados ao modo gráfico.

**ClrEol [CRT] (P)**

Apaga todos os caracteres desde a posição do cursor até o fim da linha, mantendo a localização do cursor.

**ClrScr [CRT] (P)**

Limpa a tela e coloca o cursor no canto superior esquerdo.

**Concat(s1,...,sn : String) : String [SYS] (F)**

Concatena os strings especificados.

**Copy(s : String; início, tamanho : Integer) : String [SYS] (F)**

Retorna substring de s; começa em início e tem o tamanho especificado.

**Cos(x : <real>) : <real> [SYS] (F)**

Retorna o co-seno de x (em radianos).

**CSeg : Word [SYS] (F)**

Retorna o conteúdo do registrador CS.

**Dec(VAR x : <ordinal> [:n : LongInt]) [SYS] (P)**

Decrementa uma variável x em 1 ou em n.

**Delay(x : Word) [CRT] (P)**

Espera durante x milissegundos.

**Delete(VAR s : String; começo, tamanho : Integer) [SYS] (P)**

Elimina parte do string s; começa em início até o tamanho especificado.

**DelLine [CRT] (P)**

Elimina a linha onde está o cursor.

**DetectGraph(VAR graph\_driver, modo : Integer) [GRP] (P)**

Determina se o computador é dotado de dispositivo gráfico e qual é o **graphDriver** (driver em software), bem como o **modo** que o **InitGraph** deve usar. Automaticamente acionado por **InitGraph**; se **graph\_driver** = -2 o hardware gráfico não está presente.

**DiskFree(drive : WORD) : LongInt [DOS] (F)**

Retorna o número de bytes livres no drive especificado (drive default=0, A=1,B=2,...). Retorna -1 se o drive não estiver presente.

**DiskSize(drive : Byte) : LongInt [DOS] (F)**

Retorna o tamanho total em bytes do disco no drive especificado (drive default=0, A=1,B=2,...).

**Dispose(VAR p : <pointer>) [SYS] (P)**

Libera o espaço ocupado no heap pela variável apontada por p.

**DosExitCode : Word [DOS] (F)**

Código de retorno de um subprocesso. O byte menos significativo contém o código de retorno do processo; o byte mais significativo contém:

- 0 término normal
- 1 término com CTRL+C
- 2 término por erro de dispositivo
- 3 término por procedure **Keep**

**DosVersion : Word [DOS] (F)**

Retorna a versão do DOS. Byte menos significativo = versão principal; byte mais significativo = versão secundária.

**DrawPoly(np : Word; VAR matriz) [GRP] (P)**

Traça polígono ligando as np coordenadas contidas em matriz.

**DSEg : Word [SYS] (F)**

Retorna o conteúdo do registrador DS.

**Ellipse(x, y : Integer; angl, angF : Word; raiox, raioy : Word) [GRP] (P)**

Desenha uma elipse centrada em x,y, começando no ângulo angl e terminando em angF (a partir do eixo x); raiox e raioy dão largura e altura.

**EnvCount : Integer [DOS] (F)**

Retorna o número de strings no environment.

**EnvStr(n : Integer) : String [DOS] (F)**

Retorna o string número n do environment.

**Eof((VAR f : Text)) : Boolean [SYS] (F) - (Arquivos Texto)**

Retorna True se o fim do arquivo f for atingido. Caso f seja omitido, assume o dispositivo padrão **Input**.

**Eof(VAR f : Text) : Boolean [SYS] (F) - (Arquivos Tipados ou Não Tipados)**

Retorna True se o fim do arquivo f for atingido.

**Eoln((VAR f : Text)) : Boolean [SYS] (F)**

Retorna True se o fim de uma linha em um arquivo texto for atingido.

**Erase(VAR f : <file>) [SYS] (P)**

Elimina o arquivo f do diretório.

**Exec(path, cmdline : String) [DOS] (P)**

Carrega e executa outro programa (path) com argumentos cmdline. Deve-se chamar **SwapVectors** antes e depois de sua utilização.

**Exit [SYS] (P)**

Abandona o bloco ativo.

**Exp(x : <real>) : <real> [SYS] (F)**

Retorna e<sup>x</sup>.

**FExpand(path : PathStr) : PathStr [DOS] (F)**

Retorna em path o nome completo do arquivo que originalmente foi especificado em path. Vide tipo **PathStr** na unit DOS.

**FilePos(VAR f : File) : LongInt [SYS] (F)**

Retorna o registro ativo (não deve ser usado em arquivo TEXT).

**FileSize(VAR f : File) : LongInt [SYS] (F)**  
Retorna o tamanho do arquivo (não deve ser usado em arquivo TEXT).

**FillChar(VAR v; n : Word; c : <ordinal>) [SYS] (P)**  
Coloca n bytes de c a partir do endereço ocupado pela variável v.

**FillEllipse(x,y : Integer; angl, angF : Word; raiox, raioy : Word) [GRP] (P)**  
Desenha uma elipse centrada em x,y, começando no ângulo angl e terminando em angF (a partir do eixo x); raiox e raioy dão largura e altura. A elipse terá a cor e o padrão ativos.

**FillPoly(np : Word; VAR matriz) [GRP] (P)**  
Traça um polígono ligando as np coordenadas contidas em matriz. O polígono terá a cor e o padrão ativos.

**FindFirst(path : string; atrib : Byte; VAR s : SearchRec) [DOS] (P)**  
Localiza a primeira entrada no diretório que satisfaz o path (pode incluir \* e ?). Inclui na pesquisa arquivos com os atributos atrib. O resultado será apresentado na variável s tipo SearchRec definida na unit DOS.

**FindNext(VAR s : SearchRec) [DOS] (P)**  
Localiza o próximo arquivo que satisfaz as especificações de FindFirst. O resultado será apresentado na variável s tipo SearchRec definida na unit DOS.

**FloodFill(x, y, borda : Word) [GRP] (P)**  
Pinta uma área com a cor e o padrão estabelecidos por SetFillStyle e SetFillPattern; borda define a cor da divisa da área; x,y é um ponto dentro ou fora da área delimitada e define o que deve ser pintado.

**Flush(VAR f : Text) [SYS] (P)**  
Grava o conteúdo do buffer de um arquivo f tipo Text no disco.

**Frac(x : <real>) : <real> [SYS] (F)**  
Retorna a parte fracionária de x.

**FreeMem(VAR p : <pointer>; n : Word) [SYS] (P)**  
Liberam bytes do heap a partir do endereço indicado pelo ponteiro p; n deve ser igual ao valor previamente usado em GetMem.

**FSearch(path : PathStr; DirList : string) : PathStr [DOS] (F)**  
DirList deve conter a lista de diretórios a serem pesquisados, concatenados com ";". PathStr (vide unit DOS) retorna com o nome do arquivo encontrado, ou então "", se este não for localizado.

**FSplit(path : PathStr; VAR s1: DirString; VAR s2: NameString; VAR s3: ExtString) [DOS] (P)**  
Desdobra o nome completo de um arquivo contido em path em seus componentes: s1 drive e diretório, s2 nome do arquivo, s3 sua extensão.  
DirString = String[67] {String de acionador e diretório}  
NameString = String[8] {String de nome de arquivo}  
ExtString = String[4] {String de extensão}

**GetArcCoords(VAR coordArco: ArcCoordsType) [GRP] (P)**  
Determina as coordenadas do último comando Arc ou Ellipse.  
ArcCoordsType = RECORD  
X,Y : Integer  
XStart,YStart : Integer;  
Xend,Yend : Integer;  
END;

**GetAspectRatio(VAR aspX,aspY: Word) [GRP] (P)**  
Determina a distorção introduzida pela combinação de software/hardware gráfico relativa à representação horizontal e vertical de pontos. Ex: Comprimento\_Vertical:= aspX / aspY \* Comprimento\_Horizontal permite representar um retângulo "quadrado" na tela.

**GetBkColor : Word [GRP] (F)**  
Retorna um índice relativo à cor de fundo da palheta. Pode variar de 0 a 15, dependendo da combinação hardware/software gráfico.

**GetCBreak(VAR break : Boolean) [DOS] (P)**  
Retorna em break o estado de Ctrl+Break do DOS. Se False, esse estado só é verificado durante operações de Entrada e Saída. Se True, é verificado a cada chamada do sistema operacional.

**GetColor : Word [GRP] (F)**  
Retorna a cor do desenho ativo. Pode variar de 0 a 15, dependendo da combinação de hardware/software gráfico.

**GetDate(VAR ano, mês, dia, diasem : Word) [DOS] (F)**  
Retorna a data do sistema operacional.

**GetDefaultPalette(VAR pal: PaletteType) : PaletteType [GRP] (F)**  
Retorna em pal o tipo de palheta definida por InitGraph. Vide tipo PaletteType na unit GRAPH.

**GetDir(drive : Byte; VAR path : String) [SYS] (P)**  
Retorna o path do diretório ativo no drive indicado. (0 = drive ativo, 1 = A:, 2 = B: etc.)

**GetDriverName : String [GRP] (F)**  
Retorna o nome do driver gráfico em software.

**GetEnv(varEnv : String) : String [DOS] (F)**  
Retorna o conteúdo da variável varEnv do "environment" do DOS.

**GetFAttr(VAR f; VAR atrib : Byte) [DOS] (P)**  
Retorna o atributo de um arquivo em atrib; f deve ser do tipo File. Vide atributos nas constantes predefinidas da unit DOS.

**GetFillPattern(VAR padrão : FillPatternType) [GRP]**  
Retorna o padrão de hachura definido por SetFillPattern.

**GetFillSettings(VAR padrão\_cor : FillSettingsType) [GRP] (P)**  
Retorna o padrão de hachura e a cor definidos por SetFillPattern.  
FillSettingsType = RECORD

Pattern : Word;  
Color : Word;

END;

**GetFTime**(VAR f; VAR horário : LongInt) [DOS] (P)

Retorna a data e o horário em que um arquivo f foi criado/atualizado; **horário** pode ser decodificado por uma chamada a **UnpackTime**.

**GetGraphMode** : Integer [GRP] (F)

Retorna o modo gráfico ativo. Seu valor depende da combinação hardware/software gráfico ativa. Os valores para a placa CGA são:

| Valor | ColunaxLinhas | Palheta | Página |
|-------|---------------|---------|--------|
| 0     | 320x200       | C0      | 1      |
| 1     | 320x200       | C1      | 1      |
| 2     | 320x200       | C2      | 1      |
| 3     | 320x200       | C3      | 1      |
| 4     | 640x200       | 2 cores | 1      |

**GetImage**(x1, y1, x2, y2 : Integer; VAR bitmap) [GRP] (P)

Copia uma área da tela gráfica para um buffer **bitmap** cujo tamanho deve ser determinado por uma chamada à função **ImageSize**.

**GetIntVec**(num\_int : Byte; VAR vetor : <pointer>) [DOS] (P)

Retorna em **vetor** o endereço armazenado no vetor de interrupção **num\_int**.

**GetLineSettings**(VAR Info\_Linha: LineSettingsType) [GRP] (P)

Retorna estilo, padrão e grossura de linha definidos por **SetLineStyle**.

**LineSettingsType** = RECORD

LineStyle : Word;

Pattern : Word;

Thickness : Word;

END;

**GetMaxColor** : Word [GRP] (F)

Retorna o número máximo de cores que podem ser passadas por **SetColor**.

**GetMaxMode** : Word [GRP] (F)

Retorna o número máximo de modos apoiados pelo hardware gráfico.

**GetMaxX** : Word [GRP] (F)

Retorna o valor máximo da coordenada horizontal da tela no modo gráfico ativo.

**GetMaxY** : Word [GRP] (F)

Retorna o valor máximo da coordenada vertical da tela no modo gráfico ativo.

**GetMem**(VAR p : <pointer>; n : Word) [SYS] (P)

Reserva n bytes no heap e atribui o endereço do primeiro byte a p.

**GetModeName**(número\_do\_modos: Word) :String [GRP] (F)

Retorna o nome do modo de vídeo correspondente ao **número\_do\_modos**. Vide as constantes predefinidas na unit **GRAPH**.

**GetModeRange**(driv\_graf: Integer; VAR mínimo, máximo: Integer) [GRP] (F)

Fornece os modos gráficos **mínimo** e **máximo** válidos para o driver gráfico **driv\_graf**.

**GetPalette**(VAR palheta : PaletteType) [GRP] (P)

Retorna em **palheta** a palheta ativa e seu tamanho. Vide **PaletteType** na unit **GRAPH**.

**GetPaletteSize** : Word [GRP]

Retorna o número de cores apoiadas pelo driver gráfico instalado.

**GetPixel**(x,y : Integer) : Word [GRP] (F)

Retorna a cor do pixel na coordenada x,y.

**GetTextSettings**(VAR InfoTexto: TextSettingsType) [GRP] (P)

Retorna um registro contendo cinco campos que descrevem os atributos de tipo, direção, tamanho e bloqueamento do texto no modo gráfico. Vide constantes predefinidas em **SetTextStyle**.

**TextSettingsType** = RECORD

Font : Word;

Direction : Word;

CharSize : Word;

Horz : Word;

END;

**GetTime**(VAR hora, minuto, segundo, seg100 : Word) [DOS] (P)

Retorna o horário do sistema operacional.

**GetVerify**(VAR Verificar: Boolean) [DOS] (P)

Retorna o estado do semáforo de verificação do DOS. Se **True**, o CRC do setor escrito em disco é comparado ao CRC calculado.

**GetViewSettings**(VAR janela: ViewPortType) [GRP] (P)

Retorna os parâmetros da 'viewport' e 'clipping' ativos.

**ViewPortType** = RECORD

x1.x2.y1.y2 : Integer;

Clip : Boolean;

END;

**GetX** : Integer [GRP] (F)

Retorna a coordenada x da posição do cursor gráfico.

**GetY** : Integer [GRP] (F)

Retorna a coordenada y da posição do cursor gráfico.

**GotoXY**(x, y : Byte) [CRT] (P)

No modo texto, posiciona o cursor nas coordenadas especificadas.

**GraphDefaults** [GRP] (P)

Retorna a tela gráfica aos seus valores default.

**GraphErrorMsg**(código : Integer) : String [GRP] (F)

Retorna a mensagem de erro correspondente a **código**.

**GraphResult** : Integer [GRP] (F)

Retorna o código do erro da última operação gráfica. Vide unit **GRAPH**.

**Halt** [(código : Word)] [SYS] (P)

Encerra a execução do programa e retorna ao DOS com o código **código**, que pode ser examinado por **ERRORLEVEL** em arquivo batch.

**Hi(x : Integer|Word) : Byte [SYS] (F)**

Retorna o byte mais significativo do argumento **x** (**Integer** ou **Word**).

**HighVideo [CRT] (P)**

No modo texto ativa caracteres vídeo de alta intensidade.

**ImageSize(x1, y1, x2, y2 : Integer) : Word [GRP] (F)**

Retorna o número de bytes necessários para armazenar uma área retangular da tela.

**Inc(VAR x : <ordinal> [; n : LongInt]) [SYS] (P)**

Incrementa a variável **x** em 1 ou em **n** (se **n** for especificado).

**InitGraph(VAR grafDrv: Integer;VAR modo: Integer;path: String) [GRP] (P)**

Inicializa o sistema gráfico. Normalmente, não é necessário especificar o valor de **grafDrv** ou **modo**; **path** = " = diretório ativo.

**Insert(fonte : String; VAR s : String; início: Integer) [SYS] (P)**

Insere string **fonte** em string **s** a partir de **início**.

**InsLine [CRT] (P)**

Insere uma linha em branco na posição do cursor.

**InstallUserDriver(nome: String; autoDetectPtr: <pointer>):**

**Word [GRP] (F)**

Instala um driver de placa vídeo não padronizada; **nome** e **autoDetectPtr** são rotinas fornecidas pelo fabricante.

**InstallUserFont(nome\_arq: String) : Integer [GRP] (F)**

Permite a instalação de fonte tipográfica suprida pelo usuário. O nome do arquivo vai em **nome\_arq**. Retorna o número da fonte.

**Int(x : <real>) : <real> [SYS] (F)**

Retorna a parte inteira do argumento **x**.

**Intr(num\_int : Byte; VAR regs : Registers) [DOS] (P)**

Executa a interrupção **num\_int** (0-255) em software.

**Registers = RECORD**

**CASE Integer OF**

0 : (AX,BX,CX,DX,BP,SI,DI,DS,ES,Flags : Word);

1 : (AL,AH,BL,BH,CL,CH,DL,DH : Byte);

**END;**

**Flags** consistem em:

**FCarry** \$0001

**FParity** \$0004

**FAuxiliary** \$0010

**FZero** \$0040

**FSign** \$0080

**FOverflow** \$0800

**IOResult : Word [SYS] (F)**

Retorna o código de erro de operações de E/S quando a opção de verificação de E/S estiver inativa {\$I-}.

**Keep(codsaida : Word) [DOS] (P)**

Encerra um programa, mantendo-o na memória; não esquecer de limitar o tamanho do heap com a diretiva de compilação {\$M}.

**KeyPressed : Boolean [CRT] (F)**

Retorna **True** se alguma tecla tiver sido pressionada.

**Length(s : String) : Integer [SYS] (F)**

Retorna o número de caracteres do string **s**.

**Line(x1, y1, x2, y2 : Integer) [GRP] (P)**

Desenha uma linha de **x1,y1** até **x2,y2**.

**LineRel(dx, dy : Integer) [GRP] (P)**

Desenha uma linha do cursor gráfico (**x,y**) até (**x + dx, y + dy**).

**LineTo(x, y : Integer) [GRP] (P)**

Desenha uma linha do cursor gráfico até **x,y**.

**Ln(x : <real>) : <real> [SYS] (F)**

Retorna o logaritmo natural (base e) de **x** (e = 2.7182818285).

**Lo(x : Word|Integer) : Byte [SYS] (F)**

Retorna o byte menos significativo de **x** (**Word** ou **Integer**).

**LowVideo [CRT] (P)**

Ativa caracteres de baixa intensidade no modo texto.

**Mark(VAR p : <pointer>) [SYS] (P)**

Copia o ponteiro do heap em **p** para posterior uso por **Release**.

**MaxAvail : LongInt [SYS] (F)**

Retorna o maior bloco contíguo de memória disponível no heap.

**MemAvail : LongInt [SYS] (F)**

Retorna o número total de bytes disponíveis no heap.

**MkDir(path : String) [SYS] (P)**

Cria um subdiretório especificado em **path**.

**Move(VAR fonte, destino; n : Word) [SYS] (P)**

Copia **n** bytes da variável **fonte** para a variável **destino**.

**MoveRel(dx, dy : Integer) [GRP] (P)**

Desloca o cursor gráfico de (**x,y**) para (**x + dx, y + dy**).

**MoveTo(x, y : Integer) [GRP] (P)**

Desloca o cursor gráfico para (**x,y**).

**MsDos(VAR regs : Registers) [DOS] (F)**

Executa uma chamada (Call) a uma função do DOS; equivale a uma interrupção \$21. Vide tipo **Registers** em **Intr**.

**New(VAR p : <pointer>) [SYS] (P)**

Cria uma variável dinâmica no heap com o endereço **p**.

**NormVideo [CRT] (P)**

Retorna o modo vídeo ao seu valor original.

**NoSound [CRT] (P)**

Interrompe um tom iniciado por **Sound**.

**Odd(x : LongInt) : Boolean [SYS] (F)**

Retorna **True** se o argumento for ímpar.

**Ofs(x : Identificador) : Word [SYS] (F)**

Retorna o endereço (deslocamento) de **x**.

**Ord(x : <ordinal>) : LongInt [SYS] (F)**  
Retorna <ordinal> representando a ordem de x em um conjunto de valores ordinais.

**OutText(texto : String) [GRP] (P)**  
No modo gráfico, escreve na tela o string **texto** na posição do cursor.

**OutTextXY(x, y : Integer; texto : String) [GRP] (P)**  
No modo gráfico escreve na tela o string **texto** nas coordenadas x,y.

**OvrClearBuf [OVR] (P)**  
Limpa o buffer de overlay (superposição de módulos).

**OvrGetBuf : LongInt [OVR] (F)**  
Retorna o tamanho corrente do buffer de overlay.

**OvrInit(NomArq: String) [OVR] (P)**  
Inicializa o gerenciador de overlay e abre o arquivo de overlay. O código de erro é retornado na variável **OvrResult**.

**OvrInitEMS [OVR] (P)**  
Se possível, carrega os overlays em memória estendida.

**OvrSetBuf(tamanhoBuffer: LongInt) [OVR] (P)**  
Determina o tamanho do buffer de overlays.

**PackTime(VAR dt : DateTime; VAR horário : LongInt) [DOS] (P)**  
Retorna em **horário** a variável **dt** compactada. Vide **DateTime** na unit **DOS**.

**ParamCount : Word [SYS] (F)**  
Retorna o número de argumentos na linha de comando.

**ParamStr(n : Word) : String [SYS] (F)**  
Retorna o n-ésimo argumento passado na linha de comando.

**Pi : <real> [SYS] (F)**  
Retorna o valor de  $\pi$  (3.1415926535897932385).

**PieSlice(x, y : Integer; angl, angF, raio : Word) [GRP] (P)**  
Desenha e preenche um setor de um gráfico tipo torta, usando x,y como ponto central e desenhando de **angl** até **angF** medidos em relação ao eixo x.

**Pos(s1, s2 : String) : Byte [SYS] (F)**  
Retorna a posição do início do string **s1** dentro do string **s2**.

**Pred(x : <ordinal>) : <ordinal> [SYS] (F)**  
Retorna o predecessor do argumento x.

**Ptr(seg, desloc: Word) : <pointer> [SYS] (F)**  
Transforma um endereço (segmento e deslocamento) em um ponteiro.

**PutImage(x, y : Integer; VAR mapa\_bit; modo : Word) [GRP] (P)**  
Superpõe uma imagem armazenada em **mapa\_bit** na tela a partir do ponto x,y, usando a operação lógica **modo**.  
**CopyPut = 0;** {Equivale à cópia sem modificação}  
**XORPut = 1;** {Imagem é superposta com XOR na tela}  
**OrPut = 2;** {Imagem é superposta com OR na tela}  
**AndPut = 3;** {Imagem é superposta com AND na tela}  
**NotPut = 4;** {Imagem é superposta com NOT na tela}

**PutPixel(x, y : Integer; cor: Word) [GRP] (P)**  
Desenha um pixel nas coordenadas x,y com a cor indicada.

**Random [(faixa : Word)] : Real [SYS] (F)**  
Retorna um número pseudo-aleatório entre 0 e **faixa**. Se **faixa** não for especificada, retorna valor entre 0.0 e 1.0.

**Randomize [SYS] (P)**  
Inicializa o gerador de números aleatórios com o valor atual do relógio do sistema.

**Read(f, v1 [,v2,...,vn]) [SYS] (P) - (Arquivo Tipado)**  
Lê uma ou mais variáveis de um arquivo **f** de qualquer tipo menos **Text**.

**Read([VAR f : Text;] v1 [,v2,...,vn]) [SYS] (P)**  
Lê uma ou mais variáveis de um arquivo **Text f**. Se **f** não for especificado lê do dispositivo padrão **input**.

**ReadKey : Char [CRT] (F)**  
Retorna um caractere lido do teclado.

**Readln([VAR f : Text;] v1 [,v2,...,vn]) [SYS] (P)**  
Lê uma linha do arquivo **f**, posicionando o ponteiro na próxima linha.

**Rectangle(x1, y1, x2, y2 : Integer) [GRP] (P)**  
Desenha um retângulo usando as coordenadas **x1, y1** para o canto superior esquerdo e **x2, y2** para o canto inferior direito.

**RegisterBGldriver(driver : <pointer>) : Integer [GRP] (F)**  
Incorpora ao programa um driver de uma placa gráfica. Retorna o código de resultado.

**RegisterBGIfont(fonte : <pointer>) : Integer [GRP] (F)**  
Incorpora ao programa um driver de uma fonte tipográfica. Retorna o código de resultado.

**Release(VAR p : <pointer>) [SYS] (P)**  
Libera a memória heap acima do ponteiro **p** marcado pela procedure **Mark**.

**Rename(VAR f ;nomenovo : String) [SYS] (P)**  
Altera o nome do arquivo **f** (designação interna) para **nomenovo**.

**Reset(VAR f [:<file>; tam\_reg : Word]) [SYS] (P)**  
Abre um arquivo existente. **tam\_reg** assume o default de 128 bytes e só pode ser especificado se **file** for tipo **Untyped**.

**RestoreCrtMode [CRT] (P)**  
Restaura a tela ao modo em que se encontrava antes de **InitGraph**.

**Rewrite(VAR f : <file> [; tam\_reg : Word]) [SYS] (P)**  
Cria e abre um novo arquivo. **tam\_reg** assume o default de 128 bytes e só pode ser especificado se <file> for tipo **Untyped**.

**Rmdir(path : String) [SYS] (P)**  
Remove um subdiretório **path** vazio.

**Round(x : <real>) : LongInt [SYS] (F)**  
Retorna x arredondado para um número inteiro.

**RunError**([código\_erro : Word]) [SYS] (P)

Igual ao procedure **Halt** só que simula código\_erro.

**Sector**(x,y : Integer; angl, angF, largura, altura : Word) [GRP] (P)

Desenha e hachura um setor de elipse. **angl** e **angF** são medidos a partir do eixo x.

**Seek**(VAR f : <file> ; n : LongInt) [SYS] (P)

Posiciona o ponteiro de arquivo no registro n do arquivo f.

**SeekEof**(VAR f : Text) : Boolean [SYS] (F)

Retorna **True** se for fim de arquivo. Passa por cima de espaços, tabulações e fim de linha antes de reportar o status. Útil na leitura de números em arquivos texto.

**SeekEoln**(VAR f : Text) : Boolean [SYS] (F)

Retorna **True** se for fim de linha. Passa por cima de espaços e tabulações antes de reportar o status. Útil na leitura de números em arquivos texto.

**Seg**(x : Identificador) : Word [SYS] (F)

Retorna o endereço (segmento) de x.

**SetActivePage**(página : Word) [GRP] (P)

Define o número da página para instruções gráficas. Só funciona com placas EGA, VGA e Hércules.

**SetAllPalette**(VAR palheta) [GRP] (P)

Altera as cores para aquelas definidas por **palheta**, que é variável, não tipada, e cujo primeiro byte é o comprimento da palheta seguido da nova definição da mesma.

**SetAspectRatio**(aspectoX, aspectoY : Word) [GRP] (P)

Altera a relação default que mantém círculos com aparência redonda.

**SetBkColor**(cor : Word) [GRP] (P)

Altera a cor de fundo.

**SetCBreak**(break : Boolean) [DOS] (P)

Define o estado de verificação do **DOS** relativo ao estado de **Ctrl-Break**. Se **True**, verifica em todas as chamadas ao **DOS**; se **False**, só em operações de Entrada/Saída.

**SetColor**(cor : Word) [GRP] (P)

Define a cor usada para desenhos.

**SetDate**(ano, mês, dia, : Word) [DOS] (P)

Define a data corrente do sistema operacional.

**SetFAttr**(VAR f; atributo : Byte) [DOS] (P)

Define os atributos de um arquivo não aberto; f é o arquivo, **atributo** é a soma das constantes definidas na unit **DOS**.

**SetFillPattern**(padrão : FillPatternType; cor : Word) [GRP] (P)

Altera o padrão e cor usados por **Bar**, **Bar3D**, **FillPoly**, **FloodFill** e **PieSlice** para o padrão definido pelo usuário. **FillPatternType** = array[1..8] of Byte;

**SetFillStyle**(padrão, cor : Word) [GRP] (P)

Seleciona um dos padrões predefinidos para preenchimento: **EmptyFill** = 0; {Preenche a área com cor de fundo} **SolidFill** = 1; {Preenche a área com cor sólida}

**LineFill** = 2; {preenche com —}

**LtSlashFill** = 3; {Preenche com /// normais}

**SlashFill** = 4; {Preenche com /// grossos}

**BkSlashFill** = 5; {Preenche com \\\ grossos}

**LtBkSlashFill** = 6; {Preenche com \\\ normais}

**HatchFill** = 7; {Hachurado Leve}

**XHatchFill** = 8; {Hachurado pesado e cruzado}

**InterleaveFill** = 9; {Linhas intercaladas}

**WideDotFill** = 10; {Pontos espaçados}

**CloseDotFill** = 11; {Pontos próximos}

**UserFill** = 12; {Padrão definido pelo usuário}

**SetFTime**(VAR f; hora\_data : LongInt) [DOS] (P)

Define a data e hora de um arquivo aberto (f); **hora\_data** é criada por uma chamada à procedure **PackTime**.

**SetGraphBufSize**(tamanho\_buffer : Word) [DOS] (P)

Modifica o tamanho default (4K) do buffer de **InitGraph**.

**SetGraphMode**(modo : Integer) [GRP] (P)

Modifica o modo gráfico definido por **InitGraph** e limpa a tela.

**SetIntVec**(num\_int : Byte; endereço : <pointer>) [DOS] (P)

Altera o endereço do vetor da interrupção **num\_int** do PC.

**SetLineStyle**(estilo : Word; padrão : Word) [GRP] (P)

Define a largura e o estilo da linha; **estilo** pode ser:

**SolidLn** = 0; {Linha sólida}

**DottedLn** = 1; {Linha pontilhada}

**CenterLn** = 2; {Linha centrada}

**DashedLn** = 3; {Linha interrompida}

**UserBitLn** = 4; {Linha definida pelo usuário}

**NormWidth** = 1; {Largura normal}

**ThickWidth** = 3; {Grossa}

**SetPalette**(num\_cor : Word; cor : ShortInt) [GRP] (P)

Modifica o **num\_cor** da palheta para **cor**. Vide unit **DOS** para definição das constantes usadas.

**SetRGBPalette**(num\_cor, vermelho, verde, azul : Integer)

[GRP] (P)

Permite modificação das cores da palheta em placa VGA e IBM-8514.

**SetTextBuf**(VAR f : Text; VAR buf : <array> [1..tamanho : Word]) [SYS] (P)

Altera o tamanho do buffer de arquivo para **tamanho**. Se **tamanho** não for especificado, altera para **Sizeof(buf)**; **buf** pode ser qualquer tipo.

**SetTextJustify**(horiz, vert : Word) [GRP] (P)

Define bloqueamento de texto no modo gráfico.

**LeftText** = 0; {Bloqueamento à esquerda}

**CenterText** = 1; {Centraliza o texto}

**RightText** = 2; {Texto à direita}

**BottomText** = 0; {Texto em baixo}

**TopText** = 2; {Texto em cima}

**SetTextStyle**(fonte, direção, tam\_car : Word) [GRP] (P)

Altera as características do texto no modo gráfico: fonte, direção e tamanho.

**DefaultFont** = 0; {Fonte 8x8 mapeadas em bits}

**TriplexFont** = 1; {Fonte desenhada}

SmallFont = 2;  
 SansSerifFont = 3;  
 GothicFont = 4;  
 HorizDir = 0; {Da esquerda para a direita}  
 VertDir = 1; {De baixo para cima}  
 UserCharSize = 0; {Tamanho definido pelo usuário}

**SetTime(hora, minuto, segundo, seg100 : Word) [DOS] (P)**  
 Define o horário do sistema operacional.

**SetUserCharSize(multiX, divX, multiY, divY : Word) [GRP] (P)**  
 Permite variação de altura e largura de fontes tipográficas desenhadas. multiX:divX é a razão da largura e multiY:divY é a razão da altura.

**SetVerify(verifica : Boolean) [DOS] (P)**  
 Quando verifica = True altera o semáforo de verificação do DOS.

**SetViewPort(x1, y1, x2, y2 : Word; clip : Boolean) [GRP] (P)**  
 Define uma janela gráfica com as coordenadas x1, y1, x2, y2; clip determina se imagens serão "cortadas" nos limites da janela. Podem ser usadas as seguintes constantes predefinidas:  
 ClipOn = True; ClipOff = False;

**SetVisualPage(página : Word) [GRP] (P)**  
 Ativa a página gráfica no vídeo (somente com EGA, VGA e Hércules).

**SetWriteMode(modEsc : Integer) [GRP] (P)**  
 Determina como aparecerão os desenhos na tela; modEsc:  
 CopyPut = 0; {Equivale à cópia sem modificação}  
 XORPut = 1; {Imagem é superposta com XOR na tela}

**Sin(x : <real>) : <real> [SYS] (F)**  
 Retorna o seno de x.

**SizeOf(x) : Word [SYS] (F)**  
 Retorna o número de bytes ocupados por x, que pode ser o nome de uma variável ou uma especificação de tipo.

**Sound(hz : Word) [CRT] (P)**  
 Inicia um tom na frequência hz e continua até executar NoSound.

**SPtr : Word [SYS] (F)**  
 Retorna o conteúdo do registrador SP.

**Sqr(x : <número>) : <número> [SYS] (F)**  
 Retorna x<sup>2</sup>.

**Sqrt(x : <real>) : <real> [SYS] (F)**  
 Retorna a raiz quadrada de x.

**SSeg : Word [SYS] (F)**  
 Retorna o conteúdo do registrador SS.

**Str(x [:largura[:decimais]]; VAR s : String) [SYS] (P)**  
 Transforma um número em representação String; x pode ser Integer ou <real>; s passa a conter a representação String de x com a largura e casas decimais especificadas.

**Succ(x : <ordinal>) : <ordinal> [SYS] (F)**  
 Retorna o sucessor do argumento x.

**Swap(x : Integer) : Integer [SYS] (F)**  
 Interchangeia o byte mais significativo com o menos significativo de x.

**SwapVectors [DOS] (P)**  
 Troca os vetores de interrupção atuais pelos guardados na variável predefinida SaveIntXX. Deve ser chamado antes e depois de Exec.

**TextBackground(cor : Byte) [CRT] (P)**  
 Seleciona a cor de fundo no modo texto. Vide tabela em TextColor.

**TextColor(cor : Byte) [CRT] (P)**  
 Seleciona a cor do primeiro plano no modo texto.

|           |     |              |    |
|-----------|-----|--------------|----|
| Black     | 0   | DarkGray     | 8  |
| Blue      | 1   | LightBlue    | 9  |
| Green     | 2   | LightGreen   | 10 |
| Cyan      | 3   | LightCyan    | 11 |
| Red       | 4   | LightRed     | 12 |
| Magenta   | 5   | LightMagenta | 13 |
| Brown     | 6   | Yellow       | 14 |
| LightGray | 7   | White        | 15 |
| Blink     | 128 |              |    |

**TextHeight(texto : String) : Word [GRP] (F)**  
 No modo gráfico, retorna a altura (em pixels) de um string.

**TextMode(modos : Integer) [CRT] (P)**  
 Seleciona o modo texto especificado.

|         |     |
|---------|-----|
| BW40    | 0   |
| CO40    | 1   |
| BW80    | 2   |
| CO80    | 3   |
| Mono    | 7   |
| Font8x8 | 256 |

**TextWidth(texto : String) : Word [GRP] (F)**  
 No modo gráfico, retorna a largura (em pixels) de um string.

**Trunc(x : <real>) : LongInt [SYS] (F)**  
 Trunca um valor real x para um valor inteiro.

**Truncate(VAR f : <file>) [SYS] (P)**  
 Elimina todos os registros a partir da posição corrente. Não pode ser usado em arquivos Text.

**UnpackTime(horário : LongInt; VAR DT : DateTime) [DOS] (P)**  
 Transforma horário em forma compactada para um registro tipo DateTime definido em unit DOS.

**UpCase(c : Char) : Char [SYS] (F)**  
 Retorna o caractere c convertido em maiúscula.

**Val(s : String; VAR n : <número>; VAR e : Integer) [SYS] (P)**  
 Converte o string s em número n. A posição de qualquer erro no string s é reportada em e. Se e = 0, a conversão foi feita com sucesso.

**WhereX : Byte [CRT] (F)**  
 Retorna o número da coluna do cursor em relação à janela ativa.

**WhereY : Byte [CRT] (F)**  
 Retorna o número da linha do cursor em relação à janela ativa.

**Window(x1, y1, x2, y2 : Byte) [CRT] (P)**

No modo texto, define uma janela no vídeo.

**Write( [VAR f : <text>;] v1 [,v2,...,vn]) [SYS] (P)**

(Arquivos Texto)

Grava as variáveis v1..vn em arquivo texto f. A sintaxe de

v1..vn: **Nome\_Var[:tam\_mínimo[:casas\_decimais]]**

**tam\_mínimo** Determina o tamanho do campo reservado para **Nome\_Var**

**casas\_decimais** Determina o número de casas decimais no caso de variável Real. Se não especificado, assume o valor default.

**Write(f, v1[,v2,...,vn]) [SYS] (P)** (Arquivos tipados)

Grava as variáveis v1[...vn] em arquivo tipado f.

**Writeln( [VAR f : Text;;] v1 [,v2,...,vn]) [SYS] (P)**

Igual a **Write** só que insere marcador de fim de linha (CR,LF) depois de gravar as variáveis.

## Operadores

| Operador | Operação              | Tipo Operando         | Tipo Resultado        | Preced. |
|----------|-----------------------|-----------------------|-----------------------|---------|
| .        | Seleção de membro     | Record                | mesmo que o operando  | 1       |
| ()       | Parênteses            | Qualquer              | mesmo que o operando  | 1       |
| @        | Cria ponteiro         | Qualquer              | ponteiro              | 2       |
| +, -     | Sinal                 | Real, Integer         | mesmo que o operando  | 2       |
| NOT      | Negação               | Integer, Boolean      | mesmo que o operando  | 2       |
| *        | Multiplicação         | Real, Integer         | Real, Integer         | 3       |
| /        | Divisão               | Real, Integer         | Real                  | 3       |
| DIV      | Divisão inteira       | Integer               | Integer               | 3       |
| MOD      | Resto da divisão      | Integer               | Integer               | 3       |
| AND      | "E" lógico            | Integer, Boolean      | mesmo que o operando  | 3       |
| SHL      | Shift à esquerda      | Integer               | Integer               | 3       |
| SHR      | Shift à direita       | Integer               | Integer               | 3       |
| +        | Adição, Concatenação  | Real, Integer, String | Real, Integer, String | 4       |
| -        | Subtração             | Real, Integer         | Real, Integer         | 4       |
| OR       | "Ou" lógico           | Integer, Boolean      | mesmo que o operando  | 4       |
| XOR      | "Ou" Exclusivo lógico | Integer, Boolean      | mesmo que o operando  | 4       |
| =        | Igualdade             | Qualquer              | Boolean               | 5       |
| <>       | Desigualdade          | Qualquer              | Boolean               | 5       |
| >        | Maior que             | Qualquer              | Boolean               | 5       |
| <        | Menor que             | Qualquer              | Boolean               | 5       |
| >=       | Maior ou igual a      | Qualquer              | Boolean               | 5       |
| <=       | Menor ou igual a      | Qualquer              | Boolean               | 5       |

## Operadores de Conjuntos (SET)

| Operador | Operação     | Exemplo | Resultado                                     |
|----------|--------------|---------|-----------------------------------------------|
| *        | Interseção   | C:=A*B  | C contém os elementos comuns a A e B.         |
| +        | União        | C:=A+B  | C contém os elementos combinados de A e B.    |
| -        | Diferença    | C:=A-B  | C contém os elementos de A não contidos em B. |
| =        | Igualdade    | A=B     | True (verdadeiro) se A for igual a B.         |
| <>       | Desigualdade | A<>B    | True (verdadeiro) se A for diferente de B.    |
| >=       | Contém       | A>=B    | True (verdadeiro) se B for subconjunto de A.  |
| <=       | Está contido | A<=B    | True (verdadeiro) se A for subconjunto de B.  |
| IN       | Pertence a   | X IN B  | True (verdadeiro) se X for elemento de B.     |

## Units Padrão do Turbo Pascal

**System [SYS]**

Usada pelas outras units padrão, sempre linkeditada no programa (arquivo TURBO.TPL).

**Dos [DOS]**

Funções do DOS (arquivo TURBO.TPL).

**Crt [CRT]**

Rotinas que permitem acesso a recursos de hardware do PC (arquivo TURBO.TPL).

**Printer**

Declara o arquivo texto chamado **Lste** associa-o ao dispositivo LPT1 (arquivo TURBO.TPL).

**Graph [GRT]**

Núcleo gráfico. Contém rotinas gráficas. Usa unit **Crt** (arquivo GRAPH.TPU).

**Graph3**

Unit necessária para a compilação de programas das versões anteriores que usavam rotinas gráficas contidas em GRAPH.BIN (arquivo GRAPH3.TPU).

**Turbo3**

Unit necessária para a compilação de programas das versões anteriores. Usa unit **Crt** (arquivo TURBO3.TPU).

**Overlay [OVR]**

Permite o uso de programas maiores do que a memória. Troca partes do programa entre o disco e a memória (arquivo TURBO.TPL)

Obs.: São incorporadas ao programa com a cláusula **Uses**.

## Cabeçalho do Programa

**[ PROGRAM nome\_programa [(parâmetro,...)]]**

Identifica o programa (é opcional no Turbo Pascal).

## Cláusula Uses

**USES nome\_unit,...;** Identifica as bibliotecas (units) usadas pelo programa. Esta cláusula deve ser colocada logo depois do cabeçalho do programa.

Units padrão: **Crt**, **Dos**, **Graph**, **Graph3**, **Overlay**, **Printer**, **System** e **Turbo3**.

Units programadas pelo usuário, devendo seguir o seguinte formato:

**cabeçalho; interface implementation inicialização**

**cabeçalho**  
**interface**

Identificador **unit**.

Declara as constantes, tipos, variáveis, procedimentos e funções que são acessíveis externamente.

**implementation**  
**inicialização**

Partes não acessíveis externamente. Consiste da palavra **END** ou código de inicialização da **unit**.

## Área de Declarações

**CONST** *constante1* [:*Tipo*] = *valor*; *constante2* = *valor*;...;  
Declara a constante (pode ser uma expressão reduzida a uma constante) que será usada no programa. O valor atribuído à constante determina o tipo da mesma. O optativo **Tipo** define uma constante tipada (typed constant).

**LABEL** *rótulo*,...;  
Identifica os rótulos a serem referenciados no comando GOTO.

**TYPE** *Tipo1* = *definição\_Tipo*; *Tipo2* = *definição\_Tipo*;...;  
Declara novos tipos a partir de outros já existentes.

**VAR** *var1*,... : *Tipo1* [**ABSOLUTE** *endereço ou variável*];  
*varX*,... : *Tipo2* [**ABSOLUTE** *endereço ou variável*];...  
Declara as variáveis a serem usadas no programa.  
**ABSOLUTE** Manda o compilador alocar a variável em um endereço específico ou no mesmo endereço de outra variável

Obs. Os elementos podem ser colocados em qualquer ordem.

## Área de Definição de Procedures e Funções

**PROCEDURE** *nome* [(*par*)]; [**INTERRUPT**]; [*bloco*]  
[**EXTERNAL**]; [**FORWARD**]; [**INLINE**()];  
Define um procedimento (não retorna valor).

**FUNCTION** *nome* [(*par*):*Tipo*]; [**INTERRUPT**]; [*bloco*]  
[**EXTERNAL**]; [**FORWARD**]; [**INLINE**()];  
Define uma função (retorna valor).

Opções:

*par* [VAR] *var1*,... : *Tipo1*; [VAR] *varX*,... : *Tipo2*,...

A palavra reservada **VAR** identifica as variáveis que são passadas como endereços ao procedimento ou função. Permite alteração de seu valor. A omissão de **VAR** indica que o procedimento/função recebe apenas uma cópia do valor. A variável permanece incólume.

**INTERRUPT** Declara rotina de atendimento de interrupções.

*bloco* Bloco de comandos entre **BEGIN** e **END**; inexistente nos casos de **EXTERNAL**, **FORWARD** e **INLINE**

**EXTERNAL** Colocado após o cabeçalho da definição, indica um módulo externo em binário. Exige diretiva de compilação {**\$L arquivo**}.

**FORWARD** Colocado após o cabeçalho da definição, indica que o procedimento/função será declarado posteriormente. Permite referências anteriores ao mesmo.

**INLINE()** Insere sequência de instruções, em linguagem de máquina (listadas entre parênteses), no local de chamada da função.

## Tipos de Dados

### Tipos Simples

#### Inteiros

| Tipo     | Intervalo de Valores      | Tamanho em Bytes |
|----------|---------------------------|------------------|
| Byte     | 0 .. 255                  | 1                |
| ShortInt | -128 .. 127               | 1                |
| Integer  | -32768 .. 32767           | 2                |
| Word     | 0 .. 65535                | 2                |
| LongInt  | -2147483648 .. 2147483647 | 4                |

#### Reais

| Tipo     | Intervalo de Valores                           | Dígitos Significativos | Tamanho |
|----------|------------------------------------------------|------------------------|---------|
| Real     | $2.9 \cdot 10^{-39}$ - $1.7 \cdot 10^{38}$     | 11 - 12                | 6       |
| Single   | $1.5 \cdot 10^{-45}$ - $3.4 \cdot 10^{38}$     | 7 - 8                  | 4       |
| Double   | $5.0 \cdot 10^{-324}$ - $1.7 \cdot 10^{308}$   | 15 - 16                | 8       |
| Extended | $3.4 \cdot 10^{-4932}$ - $1.1 \cdot 10^{4932}$ | 19 - 20                | 10      |
| Comp     | $-2^{63} + 1$ - $2^{63} - 1$                   | 19 - 20                | 8       |

Os tipos **Single**, **Double**, **Extended** e **Comp** exigem o **80x87**, além da diretiva de compilação **{\$N+}**. Alternativamente, emula-se o código do **80x87** com as diretivas **{\$E+}** e **{\$N+}**. A desvantagem da emulação é o aumento do tempo de execução.

#### Tipo Boolean

Assume dois valores predefinidos, **true** (verdadeiro) e **false** (falso), representados internamente por 1 e 0. Este tipo ocupa 1 Byte.

#### Tipo Char

Caracteres ordenados de acordo com o conjunto ASCII estendido (0-255); normalmente representados entre apóstrofes. Este tipo ocupa 1 Byte.

#### Tipo Enumerado

Conjunto ordenado de identificadores cujo primeiro elemento tem a ordem 0.

Exemplo:

professor = (assistente, adjunto, pleno), o tipo da variável adjunto é professor e sua ordem é 1.

#### Tipo Subconjunto

Um subconjunto de um tipo ordinal.

Exemplos: 0..123; a..k; -32..7200.

#### Tipo String

Cadeia de até 255 caracteres. Pode-se visualizar um string como uma matriz de char.

## Tipos Estruturados

### ARRAY [a1..aN] OF Tipo

Estrutura matricial, composta de um número fixo de elementos do mesmo tipo, com índices variando de a1 até aN. A variável de índice é do tipo simples não real.

### FILE [OF Tipo]

Estrutura de arquivo, com registros do **Tipo** especificado.

**Tipo** pode ser **Text**, **typed** (qualquer tipo do Turbo Pascal) ou **untyped** que é indicado pela omissão de [OF Tipo].

Ex: 1) **TYPE** Lista: **FILE OF** Ficha;

{vide definição de Ficha abaixo}

2) Nomes: **FILE OF Text**; {Arquivo de char organizado em linhas}.

### RECORD lista\_de\_campos END

Estrutura de campos, composta de um número fixo de variáveis que podem ser de tipos diferentes. Permite um campo variante, conforme o Exemplo 2 abaixo (cada variante ocupa o mesmo espaço de memória). Exemplos:

1) **TYPE** Ficha = **RECORD**

Nome : **STRING**[35];

Altura, Peso : **Real**;

Idade : **Integer**;

Sexo : TipoSexo;

**END**;

2) **TYPE** Cidadania = **RECORD**

Nome, Sobrenome : **String**[30];

Nascimento : **Data**;

**CASE** cidadão : **Boolean OF**

true : (local : **string**[40]);

false : (país : **string**[20]; {variante 1}

porto : **string**[20]; {variante 2}

data : **Data**;

saída : **Data**);

**END**

### SET OF Tipo

Estrutura consistindo do conjunto de valores ordinais de um tipo.

Exemplo: **TYPE** ASCII = **SET OF** 0..127;

Letra = **SET OF** 'A'..'Z';

Vogal = **SET OF** ('a','e','i','o','u');

### Tipo Ponteiro

Um ponteiro (endereço) de um tipo específico de variável dinâmica localizada no heap. Exemplo: **TYPE** NomePtr = ^Nome.

### Tipo Procedure

Permite variáveis contendo ponteiro a funções ou procedures.

Exemplo: **Type** Troca = **procedure**(var a,b:integer);

### Mudança de Tipo

Uma variável declarada de um tipo pode ser transformada (type cast) em outro. É necessário que os dois tipos ocupem o mesmo espaço de memória. Exemplo:

**Var** A:integer; B:word;

**BEGIN**

A := -139;

B := word(A);

**END.**

## Constantes, Variáveis e Tipos Predefinidos

### Unit CRT

#### Variáveis

**ChecBreak** : **Boolean**

Se **True**, permite que o programa seja interrompido com **CTRL-BREAK**.

**CheckEOF** : **Boolean**

Se **True**, **CTRL-Z** gera caractere indicando fim de arquivo. Default = **False**.

**DirectVideo** : **Boolean**

Ativa/desativa o acesso direto à memória do vídeo com os comandos **Write** e **Writeln**.

**CheckSnow** : **Boolean**

Ativa/desativa a verificação de "neve" quando está armazenando caracteres diretamente na memória do vídeo.

**LastMode** : **Word**

Contém o modo do vídeo antes de **TextMode**.

**TextAttr** : **Byte**

Contém os atributos do modo texto selecionado.

**WindMin** : **Word**

Coordenada superior esquerda da janela.

**WindMax** : **Word**

Coordenada inferior direita da janela.

### Unit DOS

#### Constantes

Usadas por **GetFAttr**, **SetFAttr**, **FindFirst**, e **FindNext**

**ReadOnly** \$01

**Hidden** \$02

**SysFile** \$04

**VolumeID** \$08

**Directory** \$10

**Archive** \$20

**AnyFile** \$3F

#### Tipos

Usados por **UnpackTime**, **PackTime**, **GetFTime**, **FindFirst** e **FindNext**

**DateTime** = **RECORD**

Year, Month, Day, Hour, Min, Sec : **Integer**;

**END**;

Usados em **FindFirst** e **FindNext**

type

**SearchRecord** = **RECORD**

Fill : **Array**[1..21] of **Byte** {reservado pelo DOS}

Attr : **Byte** {atributos do arquivo}

Time : **LongInt** {data e hora compactadas}

Size : **LongInt** {Tamanho do arquivo em Bytes}

Name : **String**[12] {Nome do arquivo}

**END**;

#### Variável de Erro do DOS

**VAR** DosError: **Integer**; {Contém um dos códigos abaixo}

0 Sem erro

2 Arquivo não encontrado

3 Path não encontrado

5 Acesso não permitido

6 Handle inválido

8 Insuficiência de memória

10 Environment inválido

11 Formato inválido

18 Não existem mais arquivos

## Unit GRAPH

### Tipos

Registros usado por **GetPalette**, **GetDefaultPalette** e **SetAllPalette**:

Type

PaletteType = RECORD

Size : Byte

Colors : Array[0...MaxColors] OF ShortInt;

END;

Tipo definido de acordo com a conveniência do usuário

PointType = RECORD

X,Y : Integer;

END;

### Constantes

Usadas por **InitGraph**, **DetectGraph** e **GetModeRange**

Detect = 0; Solicita auto-deteção

CGA = 1;

MCGA = 2;

EGA = 3;

EGA64 = 4;

EGAMono = 5;

IBM8514 = 6;

HercMono = 7;

ATT400 = 8;

VGA = 9;

PC3270 = 10;

CurrentDriver = -128; Driver ativo

CGA0 = 0; 320x200 palheta 0

CGA1 = 1; 320x200 palheta 1

CGA2 = 2; 320x200 palheta 2

CGA3 = 3; 320x200 palheta 3

CGAHi = 4; 640x200 2 cores

EGALo = 0; 640x200 16 cores 4 páginas

EGAHi = 1; 640x350 16 cores 2 páginas

HercMonoHi = 0; 720x348 mono 2 páginas

VGALo = 0; 640x200 16 cores 2 páginas

VGAMed = 1; 640x350 16 cores 2 páginas

VGAHi = 2; 640x480 16 cores 1 página

Uso em **SetPalette** e **SetAllPalette**:

Black = 0;

Blue = 1;

Brown = 6;

Cyan = 3;

DarkGray = 8;

Green = 2;

LightCyan = 11;

LightBlue = 9;

LightGray = 7;

LightGreen = 10;

LightMagenta = 13;

LightRed = 12;

Magenta = 5;

Red = 4;

White = 15;

Yellow = 14;

Constante usada por **GetPalette**, **GetDefaultPalette**, **SetAllPalette** e para definição de registro tipo **PaletteType**:

MaxColor = 15

### Variáveis

Códigos de erro retornados pela variável **GraphResult**.

|                        |                                      |
|------------------------|--------------------------------------|
| grOk = 0;              | Sem erro                             |
| grNoInitGraph = -1;    | Unidade gráfica não instalada        |
| grNotDetected = -2;    | Hardware gráfico não instalado       |
| grFileNotFound = -3;   | Driver não localizado                |
| grInvalidDriver = -4;  | Arquivo driver inválido              |
| grNotLoadMem = -5;     | Memória insuficiente para driver     |
| grNoScanMem = -6;      | Memória insuficiente para scan fill  |
| grNoFloodMem = -7;     | Memória insuficiente para flood fill |
| grFontNotFound = -8;   | Fontes gráficas não localizadas      |
| grNoFonMem = -9;       | Memória insuficiente para fontes     |
| grInvalidMode = -10;   | Modo gráfico inválido para driver    |
| grError = -11;         | Erro genérico                        |
| grIOError = -12;       | Erro de entrada/saída gráfica        |
| grInvalidFont = -13;   | Arquivo de fontes inválido           |
| grInvalidFonNum = -14; | Número de fonte inválido             |

## Unit OVERLAY

### Constantes

|                      |                                               |
|----------------------|-----------------------------------------------|
| ovrOk = 0;           | {Sucesso na operação}                         |
| ovrError = -1;       | {Erro do gerenciador de overlay}              |
| ovrNotFound = -2;    | {Arquivo Overlay não encontrado}              |
| ovrNoMemory = -3;    | {Memória insuficiente para buffer de overlay} |
| ovrIOError = -4;     | {Erro de Entrada/Saída de arquivo}            |
| ovrNoEMSDriver = -5; | {Driver de memória expandida não instalado}   |
| ovrNoEMSMemory = -6; | {Falta memória EMS}                           |

## Unit SYSTEM

### Variáveis

|                      |                                                                          |
|----------------------|--------------------------------------------------------------------------|
| Input : Text;        | Arquivo padrão de entrada                                                |
| Output : Text;       | Arquivo padrão de saída                                                  |
| SaveIntXX: <pointer> | Variável de armazenamento da interrupção \$xx                            |
|                      | xx = 02, 1B, 23, 24, 34, 35, 36, 37, 38, 39, 3A, 3B, 3C, 3D, 3E, 3F e 75 |

## Caracteres Especiais

| Caractere    | Significado                                                                                                         |
|--------------|---------------------------------------------------------------------------------------------------------------------|
| #            | Prefixo que identifica o valor ordinal do caractere (0-255).                                                        |
| \$           | Prefixo que identifica um número hexadecimal. Ex: \$5A.                                                             |
| .            | 1) Fim de programa<br>2) Seleção de membro em record<br>3) programa/unit.identificador (da origem do identificador) |
| ..           | Identifica um intervalo em tipo ordinal simples. Ex. 1..5.                                                          |
| ;            | Separa declarações e indica fim de definições, cabeçalhos, blocos, etc.                                             |
| [] ou (..)   | Referência a um elemento de um array.                                                                               |
| ^            | Identifica um pointer ou um caractere de controle.                                                                  |
| { } ou ( * ) | Delimita um comentário dentro do programa.                                                                          |

## Comandos do Editor

### Movimentação do Cursor

|                                 |                      |
|---------------------------------|----------------------|
| Caractere à esquerda            | Ctrl+S ou ←          |
| Caractere à direita             | Ctrl+D ou →          |
| Palavra à esquerda              | Ctrl+A               |
| Palavra à direita               | Ctrl+F               |
| Linha acima                     | Ctrl+E ou ↑          |
| Linha abaixo                    | Ctrl+X ou ↓          |
| Rola texto uma linha para baixo | Ctrl+W               |
| Rola texto uma linha para cima  | Ctrl+Z               |
| Página acima                    | Ctrl+R ou PGUP       |
| Página abaixo                   | Ctrl+C ou PGDN       |
| Início da linha                 | Ctrl+QS ou HOME      |
| Fim da linha                    | Ctrl+QD ou END       |
| Topo da janela                  | Ctrl+QE ou Ctrl+HOME |
| Última linha da janela          | Ctrl+QX ou Ctrl+END  |
| Início do arquivo               | Ctrl+QR ou Ctrl+PGUP |
| Fim do arquivo                  | Ctrl+QC ou Ctrl+PGDN |
| Início do bloco                 | Ctrl+QB              |
| Fim do bloco                    | Ctrl+QK              |
| Última posição do cursor        | Ctrl+QP              |
| Última posição de erro          | Ctrl+QW              |
| Tabulação                       | Ctrl+I ou TAB        |
| Liga/desliga auto-endentação    | Ctrl+QI              |

### Comandos de Inserção e Eliminação

|                              |                     |
|------------------------------|---------------------|
| Ativa/desativa a inserção    | Ctrl+V ou INS       |
| Insere linha                 | Ctrl+N              |
| Elimina a linha corrente     | Ctrl+Y              |
| Elimina até o fim da linha   | Ctrl+QY             |
| Elimina caractere à esquerda | Ctrl+H ou BACKSPACE |
| Elimina caractere            | Ctrl+G ou DEL       |
| Elimina palavra à direita    | Ctrl+T              |

### Manipulação de Blocos

|                              |               |
|------------------------------|---------------|
| Marca início do bloco        | Ctrl+KB ou F7 |
| Marca fim do bloco           | Ctrl+KK ou F8 |
| Marca uma palavra            | Ctrl+KT       |
| Copia bloco                  | Ctrl+KC       |
| Transfere bloco              | Ctrl+KV       |
| Elimina bloco                | Ctrl+KY       |
| Lê bloco do disco            | Ctrl+KR       |
| Grava bloco no disco         | Ctrl+KW       |
| Mostra/esconde bloco         | Ctrl+KH       |
| Imprime bloco                | Ctrl+KP       |
| Endenta bloco                | Ctrl+KI       |
| Remove a endentação do bloco | Ctrl+KU       |

### Outros Comandos

|                                           |                    |
|-------------------------------------------|--------------------|
| Finaliza a edição, sem gravar             | Ctrl+KD ou Ctrl+KQ |
| Grava e continua a edição                 | Ctrl+KS ou F2      |
| Carrega um novo arquivo                   | F3                 |
| Tabulação                                 | Ctrl+I ou TAB      |
| Ativa/desativa tabulação                  | Ctrl+OT ou Ctrl+QT |
| Liga/desliga auto-endentação              | Ctrl+OI ou Ctrl+QI |
| Recupera uma linha que foi apagada        | Ctrl+QL            |
| Define um marcador n na posição do cursor | Ctrl+Kn            |
| Posiciona o cursor em um marcador n       | Ctrl+Qn            |

Pesquisa um string Ctrl+QF

#### Opções de procura

- B Pesquisa para trás.
- L Pesquisa localmente.
- N Não pede confirmação.
- G Texto inteiro.
- U Maiúsculas e minúsculas iguais.
- n Procura a n-ésima ocorrência.
- W Somente palavras inteiras.

Pesquisa e substitui string

|                                                 |         |
|-------------------------------------------------|---------|
| (opções do Ctrl+QF)                             | Ctrl+QA |
| Repete a última pesquisa                        | Ctrl+L  |
| Prefixo de caractere de controle                | Ctrl+P  |
| Interrompe a operação                           | Ctrl+U  |
| Restaura a mensagem de erro                     | Ctrl+QW |
| Insere diretivas de compilação no começo        | Ctrl+OO |
| Localiza o complemento de (.,[, {, <, ", ' ou ( | Ctrl+QI |
| Localiza o complemento de ), ], >, ", ' ou *)   | Ctrl+QJ |
| Otimiza o uso de espaços e tabulação            | Ctrl+OF |
| Tecla Backspace retrocede espaço ou tabulação   | Ctrl+OU |
| Ajuda relativa ao comando sob o cursor          | Ctrl+F1 |
| Aciona o menu principal                         | F10     |

### Hotkeys (Teclas de Ação Imediata)

|            |                                                                                        |
|------------|----------------------------------------------------------------------------------------|
| F1         | Mostra a tela de help contextual. Pressione F1 novamente para chamar o índice do help. |
| F2         | Grava o arquivo ativo que está no editor.                                              |
| F3         | Permite carregar um novo arquivo.                                                      |
| F4         | Executa o programa até a posição atual do cursor.                                      |
| F5         | Aproxima ou afasta (zoom) a janela ativa.                                              |
| F6         | Alterna a janela ativa (editor ou mensagens).                                          |
| F7         | Executa, passo a passo, (trace) entrando em sub-rotinas/funções.                       |
| F8         | Executa, passo a passo, não entrando em sub-rotinas/funções.                           |
| F9         | Executa um arquivo "make".                                                             |
| F10        | Chaveia entre janela ativa e menus.                                                    |
| Alt+F1     | Volta a mostrar a última tela de help referenciada.                                    |
| Alt+F3     | Recarrega um arquivo utilizado recentemente.                                           |
| Alt+F5     | Mostra a tela gerada pelo programa.                                                    |
| Alt+F6     | Torna ativa a próxima janela.                                                          |
| Alt+F9     | Compila o programa ativo.                                                              |
| Alt+B      | Entra no menu Break/Watch.                                                             |
| Alt+C      | Entra no menu Compile.                                                                 |
| Alt+D      | Entra no menu Debug.                                                                   |
| Alt+E      | Entra no Editor.                                                                       |
| Alt+F      | Entra no menu File.                                                                    |
| Alt+O      | Entra no menu Options.                                                                 |
| Alt+R      | Entra no menu Run.                                                                     |
| Alt+X      | Encerra o Turbo Pascal e retorna ao DOS.                                               |
| Ctrl+F1    | Mostra o help relativo ao item onde está o cursor.                                     |
| Ctrl+F2    | Encerra uma sessão de depuração.                                                       |
| Ctrl+F3    | Mostra a pilha de chamadas de rotinas.                                                 |
| Ctrl+F4    | Avalia ou modifica uma variável.                                                       |
| Ctrl+F7    | Adiciona uma expressão à janela Watch.                                                 |
| Ctrl+F8    | Chaveia a interrupção de execução (Breakpoint).                                        |
| Ctrl+F9    | Executa o programa.                                                                    |
| Ctrl+Break | Interrompe a execução do programa.                                                     |
| Esc        | Cancela uma opção do menu.                                                             |
| Shift+F10  | Mostra a versão corrente do Turbo Pascal.                                              |

## Dispositivos do Turbo Pascal

|             |                                                   |
|-------------|---------------------------------------------------|
| CON         | Console (teclado para entrada, vídeo para saída). |
| LPT1 ou PRN | Primeira impressora no sistema.                   |
| LPT2, LPT3  | Outras impressoras conectadas.                    |
| COM1 ou AUX | Primeira porta de comunicação.                    |
| COM2        | Segunda porta de comunicação.                     |
| NUL         | Dispositivo nulo. Cancela a entrada ou a saída.   |

São tratados exatamente como se fossem um arquivo, permitindo todas as operações válidas em relação a arquivos.

## Regras Para Nomes de Identificadores

Os nomes de identificadores podem ser compostos por uma letra ou sublinhado (" \_ ") seguido de qualquer combinação de letras, dígitos ou " \_ ". Somente os primeiros 63 caracteres são significativos. Letras minúsculas e maiúsculas são consideradas iguais. A redefinição de um identificador padrão (**Write**, **Ord**, etc.) impede o uso do procedimento, função ou constante normalmente associado a ele. As referências a um campo pertencente a um **record** têm a forma *registro.campo*. O uso do comando **WITH** dispensa o uso do prefixo *registro*.

## Palavras Reservadas

|             |                |         |
|-------------|----------------|---------|
| ABSOLUTE    | GOTO           | REPEAT  |
| AND         | IF             | SET     |
| ARRAY       | IMPLEMENTATION | SHL     |
| BEGIN       | IN             | SHR     |
| CASE        | INLINE         | STRING  |
| CONST       | INTERFACE      | THEN    |
| CONSTRUCTOR | INTERRUPT      | TO      |
| DESTRUCTOR  | LABEL          | TYPE    |
| DIV         | MOD            | UNIT    |
| DO          | NIL            | UNTIL   |
| DOWNT0      | NOT            | USES    |
| ELSE        | OBJECT         | VAR     |
| END         | OF             | VIRTUAL |
| EXTERNAL    | OR             | WHILE   |
| FILE        | PACKED         | WITH    |
| FOR         | PROCEDURE      | XOR     |
| FORWARD     | PROGRAM        |         |
| FUNCTION    | RECORD         |         |

Obs.: As palavras reservadas não podem ser usadas como nomes de identificadores.

## Notação Sintática Utilizada Neste Guia

|                |                                                                                                      |
|----------------|------------------------------------------------------------------------------------------------------|
| [ ]            | Indica as partes opcionais de um comando.                                                            |
| ...            | Indica repetição do último elemento.                                                                 |
| <b>negrito</b> | Palavras reservadas e constantes pré-definidas.                                                      |
| <b>comando</b> | Indica um ou mais comandos separados entre si por ";" e delimitados por <b>BEGIN</b> e <b>END</b> .  |
| <real>         | Qualquer tipo real. Vide página 23.                                                                  |
| <ordinal>      | Qualquer tipo ordinal (não real).                                                                    |
| <file>         | Qualquer tipo file, tipado ou não.                                                                   |
| (F)            | Identifica uma função predefinida.                                                                   |
| (P)            | Identifica um procedimento predefinido.                                                              |
| [SYS]          | Indica que o procedimento encontra-se na unit <b>SYSTEM</b> (ver quadro Units Padrão na página 21).  |
| [DOS]          | Indica que o procedimento encontra-se na unit <b>DOS</b> (ver quadro Units Padrão na página 21).     |
| [CRT]          | Indica que o procedimento encontra-se na unit <b>CRT</b> (ver quadro Units Padrão na página 21).     |
| [GRP]          | Indica que o procedimento encontra-se na unit <b>GRAPH</b> (ver quadro Units Padrão na página 21).   |
| [OVR]          | Indica que o procedimento encontra-se na unit <b>OVERLAY</b> (ver quadro Units Padrão na página 21). |

## Espaço para suas anotações