

**Universidade de São Paulo  
USP  
São Carlos**

**Descobrimdo  
o ambiente do  
TURBO PASCAL  
(Versão 7.0)**

**Marcelo Augusto Santos Turine  
José Carlos Maldonado  
Rosely Sanches**

**Agosto de 1995**

## 1. Introdução

Este trabalho é destinado a estudantes de graduação interessados em familiarizar-se com a versão 7.0 do ambiente do Turbo Pascal da Borland. A linguagem Pascal recebeu este nome em homenagem ao filósofo e matemático do século XVII, Blaise Pascal. Esta linguagem foi desenvolvida por volta de 1968, por Niklaus Wirth, no Instituto de Informática da ETH (Eidgenössische Technische Hochschule) em Zurique, na Suíça.

A criação da linguagem Pascal teve por objetivo a obtenção de uma linguagem simples, capaz de incentivar a produção de programas claros e facilmente legíveis, favorecendo a utilização de boas técnicas de programação. Suas características foram criteriosamente escolhidas visando permitir a obtenção de programas confiáveis, modularizados e bem estruturados, processáveis por compiladores compactos, eficientes e econômicos [Farrer, 1985; Wood, 1987].

Basicamente, um programa é uma seqüência de comandos que descrevem as ações que você deseja que o computador realize, ou seja, um programa consiste na tradução de um algoritmo em uma forma inteligível para a máquina, o que é feito com o uso de uma linguagem de programação. Um programa descrito/especificado em uma linguagem de alto nível é denominado de *programa fonte*. Um *algoritmo* é uma seqüência ordenada de passos executáveis e precisamente definidos, que manipulam um conjunto de informações a fim de obter um resultado. Estas informações são denominadas de dados [Cereda & Maldonado, 1987].

Para que um programa possa ser executado é necessário cumprir algumas etapas (ver Figura 1):

◆ **Compilação:** Esta etapa traduz o programa fonte para o programa objeto, que são arquivos de código objeto com a extensão “.OBJ”. Esses arquivos contêm um código binário que não pode ser entendido na exibição e nem na impressão.

◆ **Composição:** Ao ser chamado, o editor de ligação (*linker*) toma um ou mais arquivos “.OBJ” e processa-os juntos para produzir um único arquivo executável com a extensão “.EXE”. O editor de ligação pode também processar o código proveniente de bibliotecas de rotinas pré-compiladas fornecidas pelo Pascal.

◆ **Execução:** Após criar o arquivo executável, seu programa já pode ser executado. Depois do comando de execução (o nome do arquivo executável) entre com os dados exigidos pelo programa (se necessários) e espere os resultados.

Os programas fontes em Pascal devem sempre ser identificados através de um nome, que deverá conter, obrigatoriamente, a extensão “.PAS”. As etapas de compilação e composição descritas acima gerarão dois novos arquivos identificados pelo mesmo nome atribuído ao arquivo programa fonte, porém diferenciados pelas extensões “.OBJ” e “.EXE”, respectivamente (por exemplo: *primeiro.pas*, *primeiro.obj*, *primeiro.exe*).

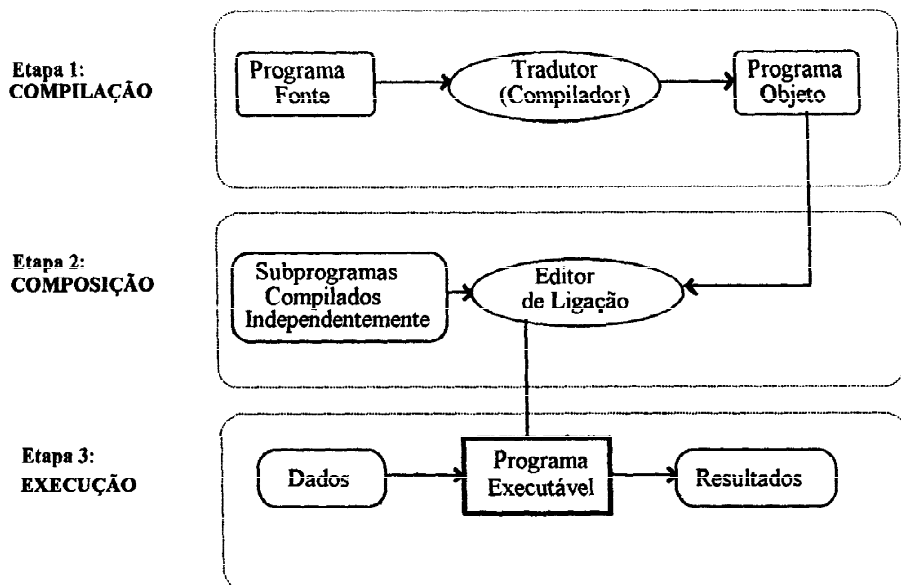


Figura 1: Etapas de um processamento.

O objetivo deste trabalho é apresentar as características gerais do ambiente do Turbo Pascal da Borland, sem entrar em detalhes ou particularidades. Na seção 2 é apresentado o ambiente do Turbo Pascal, e a seguir é criado um pequeno programa Pascal para exemplificar as etapas descritas acima. As etapas gerais de processamento de um programa, bem como, algumas situações de erros que podem ocorrer nessas etapas e os possíveis procedimentos de correção destes erros são apresentados nas seções 3 e 4. Por último, são apresentadas as conclusões do trabalho.

## 2. Ambiente do Turbo Pascal

Nesta seção, inicialmente, são apresentados os passos necessários para entrar no ambiente do Turbo Pascal (versão 7.0). É importante ressaltar que tais passos são válidos para os microcomputadores configurados no Centro de Informática de São Carlos (CISC — USP, São Carlos). Observe que estes passos devem ser adaptados de acordo com a configuração do ambiente que o usuário trabalha.

Após ligar o microcomputador, siga os passos abaixo para entrar no ambiente do Turbo Pascal a partir do sistema operacional DOS (*Disk Operating System*):

1. Entre no diretório "D". Para isto, digite "D: <Enter>" após o "prompt" do DOS.

C:\>D: <Enter>      ⇒      D:\>

2. Entre no diretório que contém os arquivos necessários ao Pascal. Por exemplo, se o Pascal está instalado no diretório "Pascal" digite "cd Pascal <Enter>".

D:\>cd Pascal <Enter>      ⇒      D:\PASCAL\>

Se você instalou o Pascal em outro diretório, faça o mesmo, dando o nome do diretório e o disco.

3. Uma vez no diretório do Pascal, digite o nome do arquivo executável do Pascal que contém os comandos essenciais para configurar o ambiente.

D:\PASCAL\>turbo <Enter>

Em seguida, aparecerá a janela principal do Turbo Pascal, ilustrada na Figura 2.

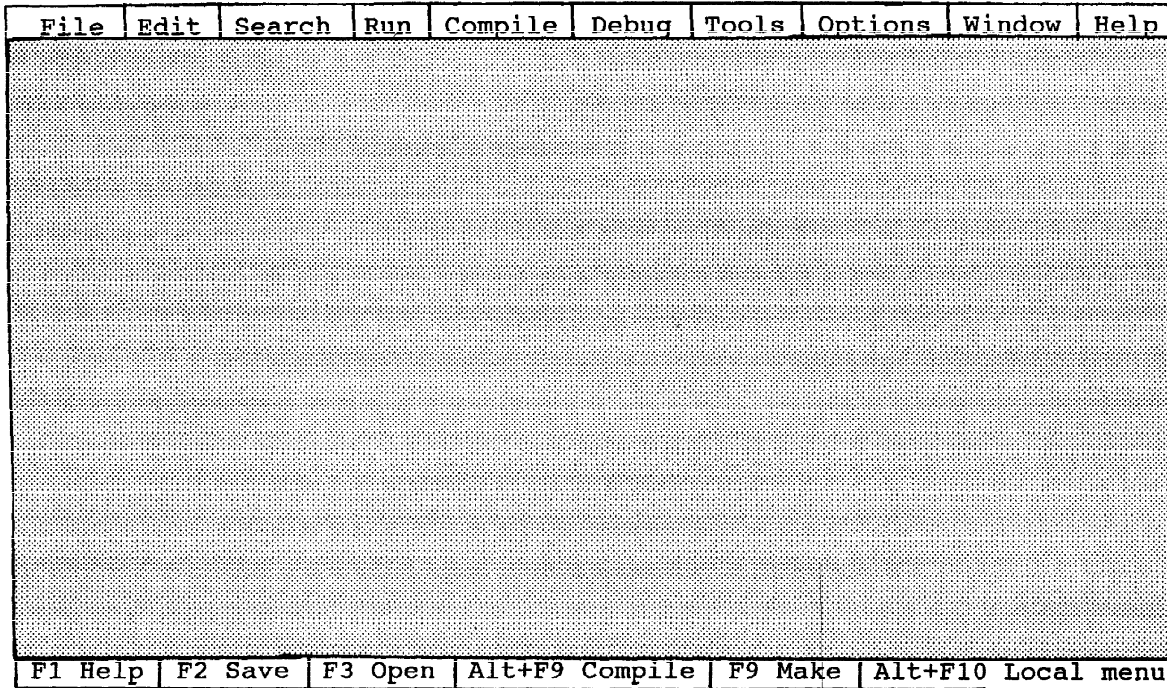


Figura 2: Janela Principal do Turbo Pascal.

Se o *mouse* estiver instalado basta selecionar, diretamente, as opções do menu principal. Caso contrário, para ativar o menu principal utiliza-se as seguintes teclas:

- A função "F10" leva você da janela de edição para o menu principal.
- "Esc" leva você de volta para a janela de edição.
- "Alt + Letra" oferece uma seleção do menu principal em qualquer instante.

A seguir, explicaremos cada uma das opções do menu principal. Caso você já esteja familiarizado com estas opções vá diretamente à seção 3.

## File (Arquivar)

New	
Open...	F3
Save	F2
Save as...	
Save all	
Change dir...	
Print	
Printer setup...	
DOS shell	
Exit	Alt+X

### New (Novo):

Limpar o editor, deixar o nome do arquivo como NONAMEn.PAS (n: é o número de janelas abertas, por exemplo, NONAME00.PAS e NONAME04.PAS) e iniciar a edição de um novo arquivo.

### Open (Abrir) - F3:

Abrir um arquivo já criado anteriormente.

### Save (Salvar) - F2:

Salvar o arquivo atual da memória para o disco.

### Save As (Salvar Como):

Salvar o conteúdo do arquivo atual em um arquivo indicado pelo usuário.

### Save All (Salvar Tudo):

Salvar todos os arquivos abertos e modificados.

### Change dir (Mudar de Diretório):

Modificar o drive ou o diretório atual.

### Print (Imprimir):

Imprimir o arquivo atual.

### Printer setup (Escolher Impressora):

Selecionar a impressora padrão para imprimir um arquivo.

### DOS Shell (Ambiente do DOS):

Sair do Turbo Pascal e volta temporariamente para o Sistema Operacional (DOS). Para voltar ao Turbo digite "EXIT".

### Exit (Sair) - Alt + X:

Sair do ambiente Fortran.

## Edit (Editar)

Undo	Alt+BkSp
Redo	
Cut	Shift+Del
Copy	Ctrl+Ins
Paste	Shift+Ins
Clear	Ctrl+Del
Show clipboard	

**Undo (Inverter) - Alt + BkSp:**

Inverter o efeito de sua operação de edição mais recente.

**Redo (Refazer):**

Inverter o efeito do último comando Undo.

**Cut (Eliminar) - Shift + Del:**

Eliminar um bloco de texto selecionado na janela de edição e copiar para o *clipboard* (área de transferência).

**Copy (Copiar) - Ctrl + Ins:**

Copiar um bloco de texto selecionado para o *clipboard*.

**Paste (Inserir) - Shift + Ins:**

Inserir um bloco de texto do *clipboard* para a janela de edição.

**Clear (Limpar) - Ctrl + Del:**

Limpar o texto selecionado sem copiar para o *clipboard*.

**Show Clipboard (Mostrar Clipboard):**

Salvar a posição atual do cursor como referência para a seleção de texto.

O editor do Turbo Pascal exibe o texto "como ele se parece" na tela e permite que um apontador ou cursor seja movimentado através do texto exibido, indicando onde mudanças ou acréscimos devem ser feitos. Como a tela pode mostrar somente 24 linhas de 80 caracteres cada, o editor exibe somente uma pequena janela do arquivo por vez.

Os comandos básicos para movimentar o cursor são:

Seta para cima (↑)	Faz o cursor subir uma linha.
Seta para baixo (↓)	Faz o cursor descer uma linha.
Seta para a esquerda (←)	Faz o cursor se mover um espaço para a esquerda.

Seta para a direita (→)	Faz o cursor se mover um espaço para a direita.
Home	Faz o cursor se mover para o início da linha.
End	Faz o cursor se mover para o final da linha.
PgUp	Faz aparecer a página anterior.
PgDown	Faz aparecer a página seguinte.
Ctrl Home	Movimenta o cursor para a primeira linha do texto.
Ctrl End	Movimenta o cursor para a última linha do texto.
<CTRL-S>	Movimenta o cursor um caractere à esquerda.
<CTRL-D>	Movimenta o cursor um caractere à direita.
<CTRL-E>	Movimenta o cursor uma linha para cima.
<CTRL-X>	Movimenta o cursor uma linha para baixo.
<CTRL-A>	Movimenta o cursor para a esquerda, uma palavra inteira de cada vez.
<CTRL-F>	Movimenta o cursor para a direita, uma palavra inteira de cada vez.
<CTRL-R>	Leva o cursor uma tela para cima.
<CTRL-C>	Leva o cursor uma tela para baixo.
<ENTER>	Insere uma nova linha e move o cursor para a linha seguinte na primeira posição tabulada.

Obviamente, um editor seria de pouca utilidade se ele não fosse capaz de inserir novas informações ou apagar informações existentes. Os comandos necessários para permitir isto são apresentados a seguir:

Ins	Muda o modo de Inserir para Sobrescrever e vice-versa.
Retrocesso	Remove o caractere à esquerda do cursor e preenche o espaço com os caracteres que estão à direita do cursor.
Del ou <CTRL-G>	Remove o caractere na posição do cursor e preenche o espaço com os caracteres que estão à direita do cursor.
<CTRL-Y>	Remove uma linha inteira.
<CTRL-T>	Remove a palavra que está à direita do cursor.
<CTRL-Q><Y>	Remove do cursor até o final da linha.
<CTRL-N>	Insere nova linha após a posição do cursor.

Uma das mais poderosas características de qualquer editor é o comando de pesquisa e troca. A capacidade de pesquisar uma seqüência de caracteres e trocá-la opcionalmente por outra pode tornar mais fácil as tarefas de edição. Por exemplo, se

you desire to change all occurrences of the word "WRITE" in a program to "WRITELN", you use only one command in the Turbo editor. A list of some search and replace commands, with their respective options is presented as follows.

**<CTRL-Q> <F>** → Asks the user to provide a sequence of characters to be located and certain parameters that specify how the search should be executed. If the user does not specify any option, the search starts at the current cursor position and continues to the end of the file. Uppercase and lowercase letters are considered, for the purpose of the search, as different characters.

Options:

**B:** The search starts at the cursor position and continues to the beginning of the file.

**G:** The search extends to the entire file. If it is the only option, the search starts at the first character of the file and goes to the last. If used in combination with option B, the search starts at the end of the file and goes to the beginning.

**n:** n is any number. Finds the n-th sequence equal to the given sequence.

**U:** Eliminates the difference between uppercase and lowercase letters for the purpose of the search.

**W:** Finds only whole words. Does not recognize sequences that are part of other words, for example, the part 'write' of the word 'writeln'.

**<CTRL-Q> <A>** → Asks the user to provide a sequence of characters to be located, a sequence to substitute and certain parameters that specify how the substitution operation should be executed. If the user does not specify any option, the substitution operation starts at the current cursor position and continues to the end of the file. Uppercase and lowercase letters are considered, for the purpose of the search, as different characters. The system asks the user to confirm each substitution.

Options:

**B:** The substitution operation starts at the cursor position and continues to the beginning of the file.

**G:** The substitution operation extends to the entire file. If it is the only option, the operation starts at the first character of the file and goes to the last. If used in combination with option B, the substitution starts at the end of the file and goes to the beginning.

**n:** n is any number. Executes the substitution n times.

**N:** The system does not ask the user to confirm the substitutions.

**U:** Eliminates the difference between uppercase and lowercase letters for the purpose of the search.

**W:** Finds only whole words.



Além destes comandos existem os comandos de blocos. Eles são utilizados para trabalhar com conjuntos de textos (palavras) de uma só vez. Todos os comandos de bloco são compostos de uma seqüência de teclas, começando com <CTRL-K>. Os comandos de blocos mais utilizados são:

<CTRL-K><B>	Marca o início do bloco.
<CTRL-K><K>	Marca o fim do bloco.
<CTRL-K><Y>	Apaga o bloco marcado.
<CTRL-K><H>	Esconde o bloco marcado.
<CTRL-K><C>	Copia bloco marcado na posição atual do cursor.
<CTRL-K><V>	Move bloco marcado para a posição atual do cursor.
<CTRL-K><W>	Escreve bloco em um arquivo em disco.
<CTRL-K><R>	Lê bloco de um arquivo em disco.

## Search (Buscar)

Find... Replace... Search again
Go to line number... Show last compiler error Find error... Find procedure...

### Find (Buscar):

Buscar a ocorrência de uma palavra no arquivo.

### Replace (Trocar):

Buscar uma palavra e trocar por outra.

### Search again (Buscar novamente):

Repetir a última busca.

### Go to line number:

Mover o cursor para uma linha específica.

### Show last compiler error:

Mostrar os últimos erros de compilação.

### Find Error (Achar Erro):

Localizar um erro de execução (run-time error) a partir do endereço do erro na notação hexadecimal: XXXX:YYYY onde XXXX é segmento e YYYY o *offset*, como por exemplo, 2BE0:FFD4.

### Find Procedure (Achar o Procedimento):

Achar um procedimento ou função no seu programa mesmo que esteja em alguma *Unit*.

## Run (Executar)

Run	Ctrl+F9
Step over	F8
Trace into	F7
Go to cursor	F4
Program reset	Ctrl+F2
Parameters...	

### Run (Executar) - Ctrl + F9:

Compilar, passar no editor de ligação (*linker*) e executar o seu programa.

### Step over (Executar por passo) - F8:

Executar o próximo comando do seu programa. Os subprogramas chamados serão executados de uma só vez.

### Trace into (Executar por passo) - F7:

Executar o próximo comando do seu programa ou subprograma (procedimento ou função).

### Go to cursor (Vá para o Cursor) - F4:

Permite executar o programa a partir da posição do cursor na janela de edição.

### Program reset - Ctrl + F2:

Terminar a depuração do programa, libera a memória e fecha os arquivos abertos.

### Parameters (Parâmetros):

Especificar os parâmetros para quando o programa for compilado em memória.

## Compile (Compilar)

Compile	Alt+F9
Make	F9
Build	
Destination Memory	
Primary file...	
Clear primary file	
Information...	

### Compile (Compilar) - Alt + F9:

Compilar e passar no editor de ligação (*linker*) o programa fonte.

### Make (Fazer) - F9:

Compilar o arquivo, checando se alguma unidade (*Unit*) foi alterada desde a última compilação. Se foi alterada, será recompilada.

Observação: “Units” são unidades de código que contêm procedimentos/funções e são compiladas separadamente do programa principal. A vantagem de se usar as “Units” é que não é necessário recompilar o programa inteiro após uma pequena alteração, somente é necessário recompilar a “Unit” afetada.

### Build (Construir):

Funciona de maneira semelhante à opção Make, porém forçando a recompilação das unidades.

### Destination Memory (Destinação):

Especificar se o código executável será armazenado no disco (como um “.EXE”) ou em memória.

### Primary file (Arquivo Principal):

O mesmo que o “Main File” da versão 3.0 do Pascal. O Turbo sempre começa a compilação por este arquivo. Utiliza-se o arquivo principal quando você está trabalhando com um programa que usa várias *units* (“.TPU”, “.TPP”, “.TPW”).

### Clear primary file (Limpar Arquivo Principal):

Limpar todos os arquivos principais especificados pela opção *Primary file*.

### Information (Informação):

Apresentar informações sobre o último programa compilado, o uso de memória e o ambiente.

## Debug (Depurar)

Breakpoints...	
Call stack	Ctrl+F3
Register	
Watch	
Output	
User screen	Alt+F5
Evaluate/modify...	Ctrl+F4
Add watch...	Ctrl+F7
Add breakpoint...	

### Breakpoints:

Ligar ou desligar um ponto de parada (*breakpoint*) na linha atual.

### Call Stack (Chamar a Pilha) - Ctrl + F3:

Visualizar, durante a depuração, as chamadas de procedimentos e funções (com seus parâmetros) realizadas para levá-lo à posição atual no programa.

### Register:

Exibir os registradores da CPU. Você utiliza esta opção quando depura seu programa no modo ASM e/ou TASM.

### Watch:

Exibir o valor de uma variável ou de uma expressão quando o programa é executado.

### Output:

A janela *Output* exibe texto (não gráfico) para qualquer linha de comando do DOS e qualquer texto gerado pelo seu programa.

### User Screen - Alt + F5:

Permite visualizar a saída de seu programa no DOS. Você pode visualizar ambos texto e gráfico na janela de saída.

### Evaluate/Modify (Avaliar) - Ctrl + F4:

Permite avaliar, visualizar e modificar o valor de uma variável ou expressão na janela *Watch*.

### Add Watch - Ctrl + F7:

Permite especificar uma variável para que seu valor seja observado na janela *Watch* durante a depuração de um programa.

### Add breakpoint:

Permite especificar (inserir) pontos de parada (*breakpoints*) no seu programa.

## Tools (Ferramentas)

Messages	
Go to next	Alt+F8
Go to previous	Alt+F7
Grep	Shift+F2

### Messages:

Exibir a janela de Mensagens.

### Go to next - Alt + F8:

Marcar e exibir a próxima mensagem da lista de mensagens

### Go to previous - Alt + F7:

Marcar e exibir a mensagem anterior da lista de mensagens

### Grep - Shift + F2:

Permite modificar e remover as ferramentas instaladas por default.

## Options (Opções)

Compiler...
Memory sizes...
Linker...
Debugger...
Directories...
Tools...
Environment ▶
Open...
Save TURBO.TP
Save as...

### Compiler (Compilador):

Permite a inserção de diretivas de compilação sem escrevê-las no programa fonte. Você pode escolher qual o compilador a gerar código, que tipo de checagem de erro fazer, que tipo de informação de depuração gerada e como o processamento numérico é controlado.

**Memory sizes:**

Permite configurar os requisitos default para a memória de seu programa, como por exemplo: *Stack size*, *Low heap limit* e *High heap limit*.

**Linker:**

Permite ajustar as opções do editor de ligação (*linker*) para seu programa.

**Debugger:**

Fornece várias opções para configurar a depuração de seu programa.

**Directories (Diretórios):**

Mostra ao Turbo em qual diretório devem ser procurados e/ou gravados os arquivos (programas fontes, arquivos de inclusão, unidades e arquivos objetos).

**Tools:**

Permite adicionar, apagar ou editar programas no menu *Tools*. Você pode escolher itens do menu *Tools* para executar outro programa sem sair do Turbo Pascal.

**Environment (Ambiente):**

Permite selecionar as opções de gravação automática da configuração do Turbo Pascal e do arquivo editado, criação de cópias de segurança dos arquivos fontes, seleção da quantidade de linhas para a tela (25,43 ou 50 dependendo da placa gráfica), tamanho e visualização de janelas para *Edit*, *Watch*, Saída, etc.

**Open:**

Permite abrir um arquivo de configuração do Turbo Pascal (*compiler*, *linker*, *debugger*) que foi salvo em algum arquivo em disco com a extensão “.TP”.

**Save:**

Grava a configuração atual do Turbo Pascal para um arquivo em disco já especificado.

**Save as:**

Grava a configuração atual do Turbo Pascal para um arquivo em disco que será lido neste momento.

## Window (Janela)

File	
Cascade	
Close all	
Refresh display	
Size/Move	Ctrl+F5
Zoom	F5
Next	F6
Previous	Shift+F6
Close	Alt+F3
List...	Alt+O

### Tile:

Permite organizar todas as janelas abertas no Turbo Pascal em forma de ladrilho ( piso).

### Cascade:

Permite organizar todas as janelas abertas no Turbo Pascal em forma de cascata.

### Close All:

Fechar todas as janelas abertas.

### Refresh display:

Permite visualizar a janela se ocorreu alguma escrita accidental.

### Size/Move - Ctrl + F5:

Permite modificar o tamanho e/ou posição da janela de trabalho (ativa).

### Zoom - F5:

Permite restaurar a janela ativa para o seu tamanho máximo.

### Next - F6:

Permite percorrer todas as janelas abertas posteriores da janela ativa.

### Previous - Shift + F6:

Permite percorrer todas as janelas abertas para trás da janela ativa.

### Close - Alt + F3:

Permite escolher uma determinada janela aberta e fechá-la.

### List - Alt + O:

Apresentar em forma de lista todas as janelas abertas no ambiente do Turbo Pascal.



## Help (Ajudar)

Contents
Index                    Shift+F1
Topic search        Ctrl+F1
Previous topic    Alt+F1
Using help
Files...
Compiler directives
Reserved words
Standard units
Turbo Pascal Language
Error messages
About...

### Contents:

Exibir a tabela de conteúdo dos assuntos explorados no ambiente do Turbo Pascal.

### Index - Shift + F1:

Exibir a lista de índices do Pascal.

### Topic search - Ctrl + F1:

Exibir a informação sobre o item ou palavra chave na qual o cursor está posicionado. Por exemplo, posicionando o cursor sobre o comando WRITELN que está na janela de edição e escolhendo esta opção (Ctrl F1) aparecerá uma janela explicando a ação do comando WRITELN e a sua sintaxe, evitando, assim, de procurar no manual do Pascal o que o comando WRITELN faz.

### Previous topic - Alt + F1:

Permite retornar ao último Topic Search pesquisado.

### Using Help:

Exibir uma explicação sobre alguma opção do Help ou sobre as janelas do ambiente. Explicar como utilizar o sistema de ajuda (help) do ambiente.

### Files:

Permitir instalar ou remover arquivos de Help.

### Compiler directives:

Apresentar as várias diretivas de compilação existentes no Turbo Pascal.

### Reserved words:

Apresentar as palavras reservadas na linguagem Pascal. Estas palavras não podem ser redefinidas novamente em qualquer programa Pascal.

### Standard units:

Apresentar os padrões de units existentes no Turbo Pascal, como por exemplo: Crt, Dos, Printer, etc.

#### Turbo Pascal Language:

Apresentar os principais elementos da linguagem Pascal, como por exemplo:  
*Variables, Units, String, etc.*

#### Error messages:

Verificar as mensagens de erros geradas pelo Turbo Pascal. O Turbo Pascal gera dois tipos de mensagens de erros: mensagens de erro de compilação(*compiler error*) e de execução (*run time error*).

#### About:

Apresentar *Copyright* e informações sobre a versão do Turbo Pascal sendo utilizada.

### 3. Edição, Compilação, Composição e Execução de um Programa

Pronto, agora você já conhece as principais opções do menu principal do Turbo Pascal. Vamos praticar! Para isto, siga os passos seguintes.

1. Entrar no ambiente do Turbo Pascal.

```
D:\>
```

```
D:\>cd pascal <Enter>
```

```
D:\PASCAL>pascal <Enter>
```

2. Criar e editar um programa fonte.

- ◆ Escolher a opção **File** do menu principal e depois *New*.

- ◆ Agora digite o programa, utilizando os recursos de edição oferecidos pelo Turbo Pascal.

O programa COBAIA, mostrado a seguir, será utilizado como exemplo para ilustrar os procedimentos descritos nesta e nas próximas seções.

{Este programa calcula e imprime o fatorial de um numero N lido pelo teclado

Autor: Marcelo

Data: 26/05/1995

Programa Principal}

```
PROGRAM COBAIA;
```

```
    {Declaracao das variaveis utilizadas}
```

```
VAR FAT, M, N: INTEGER;
```

```
BEGIN
```

```
    {Leitura de N}
```

```
    Writeln('DIGITE O VALOR DE N:');
```

```
    Readln(N);
```

```
    {Calculo do Fatorial}
```

```
    FAT:= 1;
```

```
    FOR M:=2 TO N DO
```

```
        FAT:= FAT * M;
```

```
    {Imprimir o Fatorial}
```

```
    Writeln('FATORIAL DE ',N,' = ', FAT);
```

```
END.
```

3. Ao final do processo de edição vamos Salvar o programa com o nome de COBAIA.PAS (o nome COBAIA é arbitrário; somente a extensão PAS é obrigatória).

- ◆ Escolher a opção **File** do menu principal e depois *Save*.

Digite: *C:\TMP\COBAIA.PAS* <Enter>

4. Nesse momento você já pode realizar as etapas de processamento: compilação, composição e execução do programa, como mostradas abaixo.

♦ **Compilação e Composição:** Escolher a opção **COMPILE** do menu principal e depois *Compile*.

No nosso exemplo nenhum erro de compilação e de composição será assinalado, e portanto aparecerá na tela uma mensagem de *Success* e *Press any key* piscando. Porém, se algum erro for detectado após este primeiro passo, o programa fonte deverá ser retomado com a ajuda do editor, para que as correções possam ser realizadas. Após as correções, o processo de compilação deverá ser reiniciado. Caso não haja erros, passa-se à etapa de execução.

♦ **Execução:** Escolher a opção **RUN** do menu principal e depois *Run*.

Se o programa exigir a entrada de dados (no exemplo COBAIA.PAS é necessário ler o valor de *N* para o programa calcular o fatorial) eles serão solicitados neste momento. Somente após o fornecimento dos dados de entrada é que serão mostrados os resultados do processamento.

Abaixo mostramos a execução do programa COBAIA.PAS

```
DIGITE O VALOR DE N:
```

O usuário, por sua vez, digita um valor inteiro.

```
DIGITE O VALOR DE N:  
4 <Enter>
```

Para visualizar o resultado final do programa na tela digite “ALT F5” e para voltar novamente ao ambiente digite qualquer tecla. Neste exemplo, o resultado final do programa COBAIA é:

```
FATORIAL DE 4 = 24
```

Pronto, você já sabe como editar, compilar, compor (“linkedit”) e executar um programa escrito em Pascal.

#### 4. Erros

A seqüência ilustrada acima para a qual não houve ocorrência de erros nem sempre reflete a realidade. Usualmente, o usuário comete erros de codificação, de edição ou mesmo erros na solução do problema. A detecção de erros durante as etapas de processamento de um programa é um fato bastante comum.

De forma geral, podemos ter as seguintes categorias de erros:

- ♣ Erros Sintáticos.
- ♣ Erros de Composição.
- ♣ Erros de Execução.
- ♣ Erros Semântico (de Solução).

Nas seções seguintes procuramos ilustrar as situações em que ocorrem esses erros e as sugestões de procedimentos de correção a serem conduzidos pelo programador (aluno).

#### 4.1. Erros Sintáticos

Erros detectados durante a etapa de compilação são, em geral, erros relativos às normas sintáticas da linguagem (Erros Sintáticos). Durante a etapa de compilação, os *erros sintáticos* são indicados através de números e mensagens de erros resumidas que são mostradas na tela. Mensagens de erros mais detalhadas podem ser encontradas no Guia do Usuário, que acompanha o compilador do Turbo Pascal. Para facilitar a localização de um erro o cursor é posicionado na linha do programa fonte na qual provavelmente ocorreu o erro.

Como já dito, sempre que forem detectados erros na etapa de compilação, o programa fonte deve ser retomado com a ajuda do editor, para que sejam feitas as correções. Após estas correções, deve ser reiniciada a etapa de compilação.

O exemplo apresentado a seguir mostra o programa fonte COBAIA1.PAS onde foram inseridos deliberadamente alguns erros sintáticos, como por exemplo: declaração errônea de variáveis, falta de “;” e a falta do comando END no final do programa.

```
{Este programa calcula e imprime o fatorial de um numero N lido pelo teclado
Autor: Marcelo
Data: 26/05/1995
Programa Principal}
PROGRAM COBAIA1;
  {Declaracao das variaveis utilizadas}
  VAR FAT, M, N: INTOGER                                {linha 7}
  BEGIN                                                {linha 8}
    {Leitura de N}
    WRITE ('DIGITE O VALOR DE N:');
    READLN(N);
    {Calculo do Fatorial}
    FAT:= 1;
    FOR M:=2, TO N DO
      FAT:= FAT * M;
    {Imprimir o Fatorial}
    WRITELN('  ',FATORIAL DE ',N,' = ', FAT);
  END                                                    {linha 17}
```

Ao compilar o programa acima aparecerá na linha superior da janela de Edição o número e a mensagem do erro. Este exemplo contém três erros que aparecerão um de cada vez:

Error 3: Unknow identifier	(linha 7)
Error 85: “,” expected	(linha 8)
Error 10: Unexpected end of file	(linha 17)

## 4.2. Erros de Composição

Ainda que não sejam detectados erros na fase de compilação, alguns erros ainda podem surgir durante a execução das outras etapas de processamento. Erros detectados durante a etapa de composição são denominados de *erros de composição* e usualmente se relacionam a referências não resolvidas de funções ou sub-rotinas. Se, por exemplo, ao invés de SIN(X) for escrito SEN(X) em um programa fonte, para indicar a função seno, este erro deverá ser detectado durante a etapa de composição (o editor de ligação não localiza nas bibliotecas a função SEN(X)).

O exemplo apresentado a seguir mostra o programa fonte COBAIA2.PAS onde além de calcular o fatorial de N, calcula o coseno de N. Desta maneira, foram inseridos *deliberadamente alguns erros na fase de composição*.

```
{Este programa calcula e imprime o fatorial e o coseno de um numero N lido pelo
teclado
  Autor: Marcelo
  Data: 26/05/1995
  Programa Principal}
PROGRAM COBAIA2;
  {Declaracao das variaveis utilizadas}
  VAR FAT, M, N: INTEGER;
  VAR S: REAL;
  BEGIN
    {Leitura de N}
    WRITE ('DIGITE O VALOR DE N:');
    READLN(N);
    {Calculo do Fatorial}
    FAT:= 1;
    FOR M:=2, TO N DO
      FAT:= FAT * M;
    {Calculo do Coseno}
    S:=COSENO(N);
    {Imprimir o Fatorial}
    WRITELN(' ',FATORIAL DE ',N,' = ', FAT);
    {Imprimir o Coseno}
    WRITELN(' ',COSENO DE ',N,' = ', S);
  END.
```

Ao compor o programa acima se obtém o seguinte número e mensagem de erro:

Error 3: Unknow identifier (linha 18)

#### 4.3. Erros de Execução

Erros também podem ser detectados durante a fase de execução e são denominados de *erros de execução*. Estes erros fazem com que a execução dos programas seja interrompida, ao mesmo tempo em que mostra na tela mensagens indicativas (*run-time error*). Estas mensagens nem sempre são explícitas ou fáceis de interpretar. Divisões por zero, raízes quadradas de números negativos, erros de compatibilidade de tipos nas especificações de formato, *overflow* (estouro do tamanho de uma variável), acessos a arquivos não definidos são alguns exemplos comuns de causas de erros de execução.

O exemplo apresentado a seguir mostra o programa fonte COBAIA3.PAS na qual foram inseridos deliberadamente alguns erros de execução.

```
{Este programa calcula e imprime o fatorial de um numero N lido pelo teclado
Autor: Marcelo
Data: 26/05/1995
Programa Principal}
PROGRAM COBAIA3;
  {Declaracao das variaveis utilizadas}
  VAR FAT, M, N: INTEGER
  BEGIN
    {Leitura de N}
    WRITE ('DIGITE O VALOR DE N:');
    READLN(N);
    {Calculo do Fatorial}
    FAT:= 1;
    FOR M:=2, TO N DO
      FAT:= FAT * M;
    {Imprimir o Fatorial}
    WRITELN(' ',FATORIAL DE ',N,' = ', FAT);
  END.
```

Ao executar programa acima se obtém o seguinte resultado:

```
DIGITE O VALOR DE N:
8 <Enter>

Run-time error 205 at 2095:0098
Error 205: Floating Point Overflow
```

#### 4.4. Erros Semânticos

Além dos tipos de erros discutidos até agora, outros tipos de erros podem estar associados a um programa Pascal. Este é o caso, por exemplo, dos erros presentes no projeto lógico do programa (algoritmo realizado erroneamente). Esta classe de erros pode provocar resultados indesejáveis de processamento, sem que sejam emitidas mensagens indicativas, em qualquer uma das etapas deste processamento. Sua correção exige, em geral, uma revisão completa de toda a concepção algorítmica do programa, ao final da qual se devem reiniciar as atividades de processamento. Esses tipos de erros são considerados os erros mais difíceis de serem detectados pelo usuário.

O exemplo apresentado a seguir mostra o programa fonte COBAIA4.PAS onde foi inserido deliberadamente um erro de semântica. O erro semântico ocorre na inicialização da variável FAT (FAT=0 em vez de FAT=1).

```
{Este programa calcula e imprime o fatorial de um numero N lido pelo teclado
Autor: Marcelo
Data: 26/05/1995
Programa Principal}
PROGRAM COBAIA4;
  {Declaracao das variaveis utilizadas}
VAR FAT, M, N: INTEGER
BEGIN
  {Leitura de N}
  WRITE ('DIGITE O VALOR DE N:');
  READLN(N);
  {Calculo do Fatorial}
  FAT:= 0;
  FOR M:=2 TO N DO
    FAT:= FAT * M;
  {Imprimir o Fatorial}
  WRITELN(' ',FATORIAL DE ',N,' = ', FAT);
END.
```

Ao executar programa acima se obtém o resultado:

```
DIGITE O VALOR DE N:
4 <Enter>

FATORIAL DE 4 = 0
```

Caso você tenha que escrever o(s) resultado(s) do seu programa na impressora ao invés de exibí-lo(s) na tela do computador é necessário fazer algumas adaptações. Por exemplo, no programa COBAIA5 quero imprimir o fatorial de um número qualquer na



impressora. Para isto, é necessário inserir o comando “uses printer” e ao comando de impressão WRITELN o parâmetro “LST”, como é apresentado abaixo.

```
{Este programa calcula e imprime o fatorial de um numero N lido pelo teclado
Autor: Marcelo
Data: 26/05/1995
Programa Principal}
PROGRAM COBAIA5;
  {Declaracao das variaveis utilizadas}
USES PRINTER;
VAR FAT, M, N: INTEGER
BEGIN
    {Leitura de N}
    WRITELN (LST, 'DIGITE O VALOR DE N:');
    READLN(N);
    {Calculo do Fatorial}
    FAT:= 0;
    FOR M:=2 TO N DO
        FAT:= FAT * M;
        {Imprimir o Fatorial}
    WRITELN(LST, ' ', 'FATORIAL DE ',N,' = ', FAT);
END.
```

## 5. Conclusões

Este trabalho apresentou as características gerais do ambiente do Turbo Pascal da Borland, sem entrar em detalhes ou particularidades tanto do ambiente como da linguagem de programação. Enfatizamos as etapas gerais de processamento de um programa, bem como, algumas situações de erros que podem ocorrer nessas etapas e os possíveis procedimentos de correção destes erros (sintáticos, composição, execução, semântico).

Como foi dito inicialmente não pretendemos, de forma alguma, durante o decorrer do texto esgotar todos os recursos disponíveis do Turbo Pascal (versão 7.0). Depois de familiarizado com os recursos básicos apresentados neste trabalho, sugerimos ao leitor que consulte o manual de referência do Turbo Pascal da Borland [Pascal, 1992; Turbo, 1992] e a bibliografia fornecida, entre outras fontes.

## **Bibliografia**

- [Cereda & Maldonado,1987] Cereda, R.L.D. & Maldonado, J.C. *Introdução ao FORTRAN 77 para microcomputadores*. McGraw-Hill, 1987, 211p.
- [Farrer,1985] Farrer, H.; Becker, C.G.; Faria, E.C. et.al. *Programação estruturada de computadores — Pascal Estruturado*. Guanabara Dois, 1985, 139p.
- [Pascal,1992] User's Guide, Borland - Turbo Pascal, 7.0. 1992, 176p.
- [Turbo,1992] Programmer's Reference, Borland - Turbo Pascal, 7.0. 1992, 297p.
- [Wood,1987] Wood, S.. *Turbo Pascal — Guia do Usuário (inclui versões 4.0 e 5.0)*. McGraw-Hill, 1987, 237p.

*“Há, verdadeiramente, duas coisas diferentes: saber e crer que se sabe.  
A ciência consiste em saber; em crer que se sabe reside a ignorância”*

**Hipócrates**