

**DRIVE·INTERFACE**

**DDX**



**MANUAL DE INSTRUÇÕES**

SÉRIE OURO  
1 ANO GARANTIA

**RIOSOFT**  
**MSX**

RUA CONDE DE BONFIM, 340 SLB - SCLD 107  
PÇA S. PENA - TEL. 264-3725

**DDX DRIVE INTERFACE**  
**ÍNDICE**

<b>1 - INTRODUÇÃO</b> .....	<b>pág. 2</b>
<b>1.1 - DDX DRIVE</b> .....	<b>pág. 2</b>
1.1.1 - Painel Frontal .....	<b>pág. 2</b>
1.1.2 - Painel Traseiro .....	<b>pág. 3</b>
1.1.3 - Configuração .....	<b>pág. 3</b>
<b>1.2 - DDX-INTERFACE</b> .....	<b>pág. 4</b>
1.2.1 - Conexão .....	<b>pág. 4</b>
<b>1.3 - Começando a Trabalhar</b> .....	<b>pág. 5</b>
<b>2 - MANUAL DDX-DOS</b> .....	<b>pág. 5</b>
<b>2.1 - Guia do Usuário</b> .....	<b>pág. 5</b>
2.1.1 - Caracteres Especiais .....	<b>pág. 6</b>
2.1.2 - Nomes de Arquivos Ilegais .....	<b>pág. 6</b>
2.1.3 - Diretórios .....	<b>pág. 7</b>
2.1.4 - Tipos de Comandos DDX-DOS .....	<b>pág. 7</b>
2.1.5 - Opções em Comandos .....	<b>pág. 7</b>
2.1.6 - Informações Comuns à todos os comandos DDX-DOS .....	<b>pág. 8</b>
2.1.7 - Processando Baterias (BATCH) .....	<b>pág. 8</b>
2.1.8 - O arquivo AUTOEXEC.BAT .....	<b>pág. 8</b>
2.1.9 - Como criar um arquivo bateria (BATCH) .....	<b>pág. 9</b>
2.1.10 - Parâmetros Variáveis dos arquivos .BAT .....	<b>pág. 9</b>
2.1.11 - Teclas de edição e controle do DDX-DOS .....	<b>pág.10</b>
2.1.12 - Instruções para usuários de sistemas com um único drive .....	<b>pág.12</b>
2.1.13 - Erros de Disco .....	<b>pág.12</b>
<b>2.2 - GUIA DOS COMANDOS DDX-DOS</b> .....	<b>pág.13</b>
<b>2.3 - GUIA DE BASIC DE DISCO DE MSX</b> .....	<b>pág.17</b>
2.3.1 - Comandos e Mostragens .....	<b>pág.17</b>
2.3.2 - Funções .....	<b>pág.26</b>
2.3.3 - Códigos e Mensagens de Erro .....	<b>pág.28</b>

**GARANTIA MSX - SOFT**

**ESTE APARELHO TEM**  
**GARANTIA DE 365 DIAS**  
**A PARTIR DESTA DATA**

01/12/89

## Introdução:

Este manual tem por finalidade esclarecer as dúvidas do usuário quanto a instalação e aos recursos dos DDX - PERIFÉRICOS e quanto ao sistema operacional "DDX - DOS" nos micros da linha MSX. O sistema compreende:

**DDX - DRIVE:** Caixa metálica contendo uma fonte com baixo aquecimento, um acionador de disco flexível 5 1/4" com capacidade de armazenamento de 360 K Bytes formatados, com conexão e seleção externas à caixa.

**DDX - INTERFACE:** Interface controladora de acionadores de disco e um cabo de conexão.

**DDX - DOS:** Disquete contendo o sistema operacional compatível com o MSX - DOS, permitindo, assim, a completa utilização da memória RAM de seu micro. Contém, também, jogos famosos para o MSX e um programa para criação de novos discos com sistema operacional.

Para evitar possíveis contratempos, leia este manual antes de ligar seu equipamento.

### 1.1 DDX - DRIVE

Consiste em um drive padrão IBM de 40 trilhas por polegadas de dupla densidade e dupla face. Caixa confeccionada em chapa de ferro de dimensões 15x 29x 4,5cm externamente. Fonte regulada de baixo aquecimento e placas que possibilitam, externamente, a conexão e seleção do drive.

#### 1.1.1 PAINEL FRONTAL

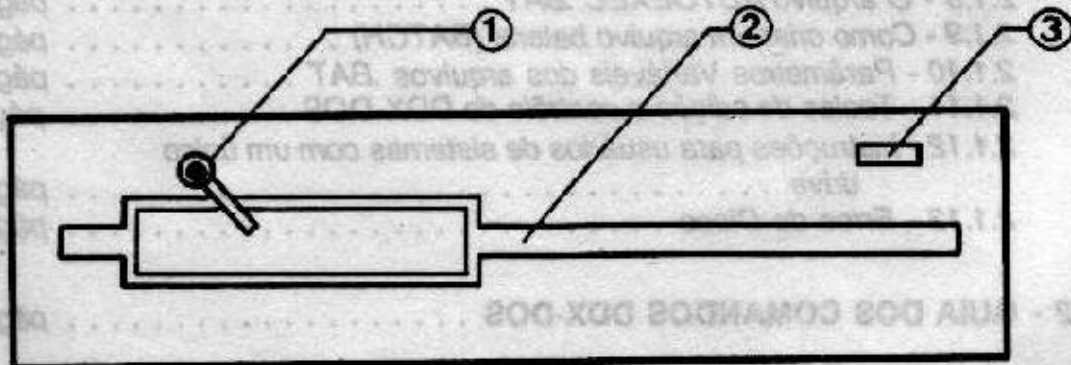


Fig. 1.1 - DDX - DRIVE

- 1 - Led indicador de seleção. Quando o drive for selecionado pelo micro, este led acenderá.
- 2 - Fenda para inserção de disco.
- 3 - Trava da porta: Esta trava deve posicionar-se verticalmente, quando o disco for inserido ou para proteção das cabeças magnéticas durante o transporte.

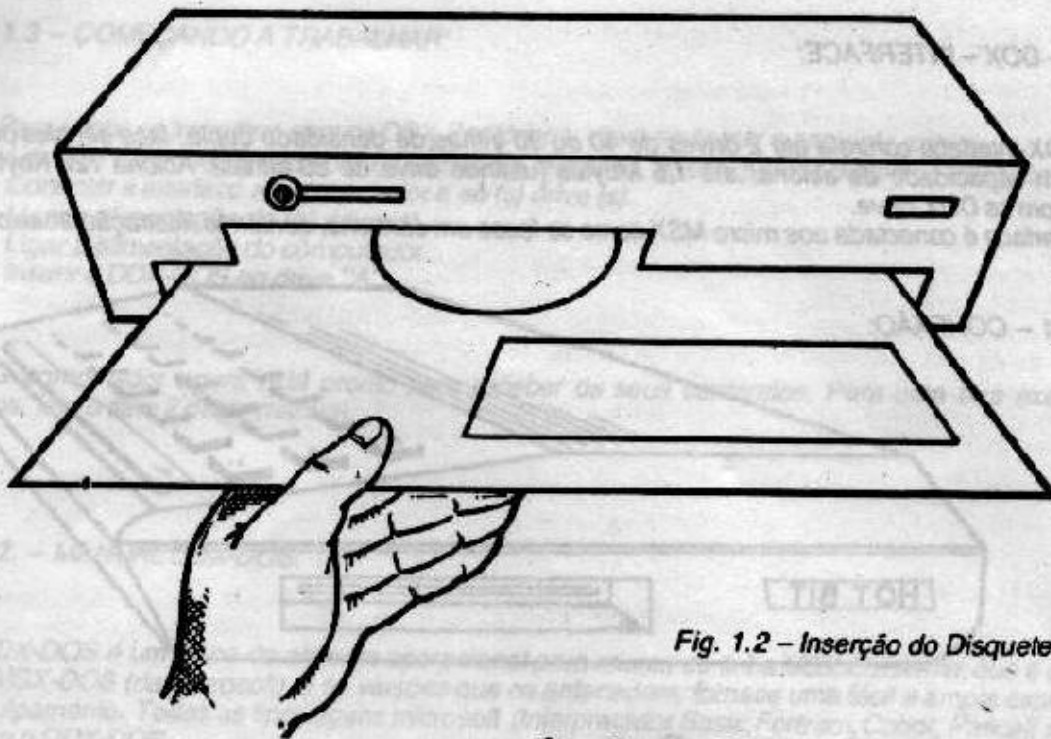


Fig. 1.2 – Inserção do Disquete.

Atenção para a inserção correta do disquete no drive. Verificar bem a posição do disquete.

#### 1.1.2 – Painel traseiro:

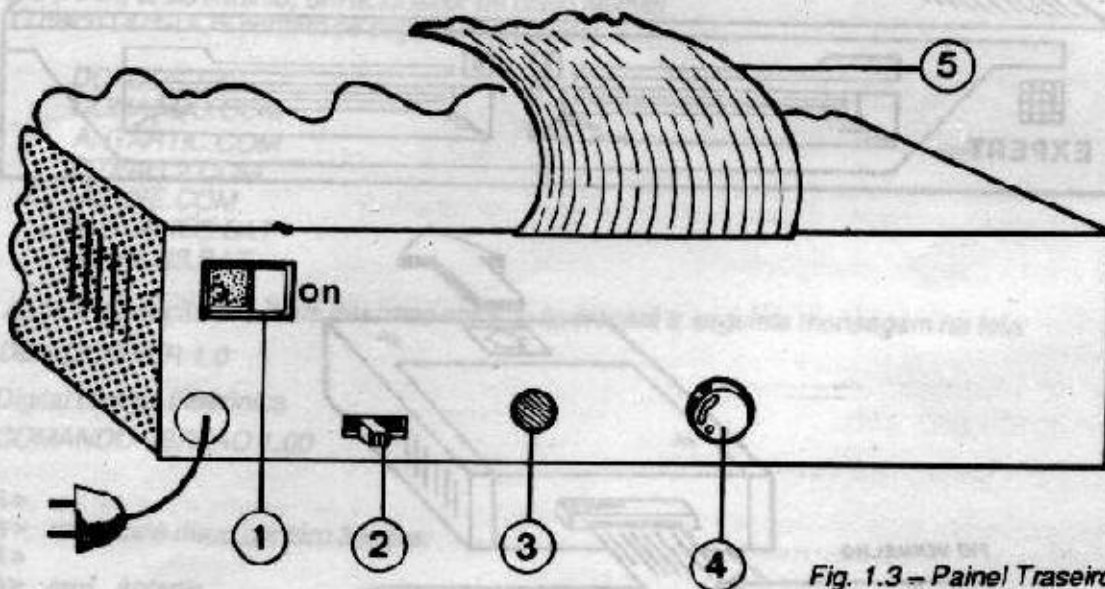


Fig. 1.3 – Painel Traseiro

- 1 – Chave ON/OFF
- 2 – Seletor de tensão: 110 ou 220volts.
- 3 – Led indicador do funcionamento da fonte.
- 4 – Porta fusível: para segurança de seu equipamento, use fusíveis de 0,5 amperes pequenos.
- 5 – Cabo de conexão (flat cable)

#### 1.1.3 – CONFIGURAÇÃO

Quando forem usados 2 drives, será necessário configurar um deles como 1 e o outro como 2 e a terminação deverá ser colocada somente no drive que estiver ligado no fim do cabo.

## 1.2 - DDX - INTERFACE:

A DDX-Interface controla até 2 drives de 40 ou 80 trilhas, de densidade dupla, face simples ou face dupla. Com capacidade de acionar até 1,5 Mbytes (usando drive de 80 trilhas). Aciona 720 Kbytes de memória com os DDX-drive.

A interface é conectada aos micro MSX como se fosse um cartucho, conforme ilustração abaixo.

### 1.2.1 - CONEXÃO:

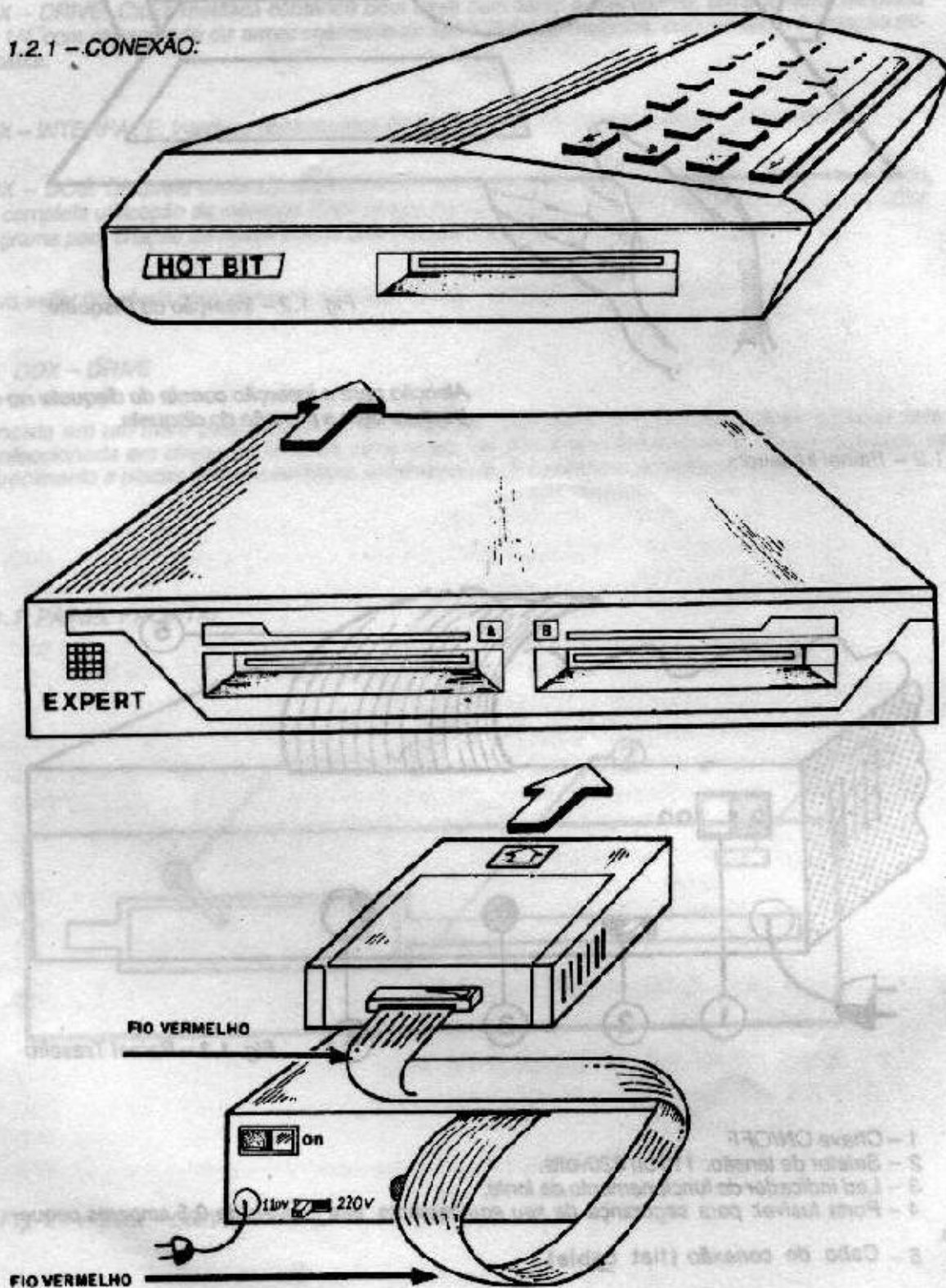


Fig. 1.4 - Conexão Interface - Micro

### 1.3 - COMEÇANDO A TRABALHAR:

Para iniciar os trabalhos com os DDX-Periféricos, deve-se seguir o seguinte roteiro:

- 1 - Ler todo o manual fornecido pelo fabricante, evitando, assim, perdas de tempo desnecessárias
- 2 - Conectar a interface ao computador e ao (s) drive (s).
- 3 - Ligar a alimentação do (s) drive (s).
- 4 - Ligar a alimentação do computador.
- 5 - Inserir o DDX-DOS no drive "A".

O computador agora está pronto para receber os seus comandos. Para uma boa execução dos mesmos, leia o item 2 deste manual.

## 2. - MANUAL DDX-DOS:

O DDX-DOS é um disco de sistema operacional para micros da linha MSX. O sistema, que é compatível com o MSX-DOS (da microsoft) e as versões que os antecedem, fornece uma fácil e ampla exploração de seu equipamento. Todas as linguagens microsoft (Interpretador Basic, Fortran, Cobol, Pascal) são utilizáveis com o DDX-DOS.

### 2.1 - GUIA DO USUÁRIO:

O sistema operacional DDX-DOS requer um micro computador da linha MSX com 64Kbytes de memória (RAM) e, no mínimo, um acionador de disco flexível.

O disco DDX-DOS contém os seguintes arquivos:

DDXDOS.SYS  
COMANDO.COM  
ANTARTIC.COM  
PITFALL2.COM  
MOUSE.COM  
AUTOEXEC.BAT  
COPIASIS.BAT

Após a execução dos itens descritos em 1.3, aparecerá a seguinte mensagem na tela:

DDX-DOS VER 1.0

Digital Design Eletrônica

COMANDO VERSÃO 1.00

- A>  
A> rem Este disco contém 3 jogos:  
A>  
A> rem Antartic  
A> rem Ajude o pinguim a dar  
A> rem a volta na Antártica  
A> rem PITFALL2  
A> rem A segunda parte da  
A> rem famosa aventura.  
A> rem MOUSE  
A> rem Ajude o rato a pegar os  
A> rem seus queijos (Fugindo dos  
A> rem gatos)  
A>  
A> rem para rodar qualquer jogo  
A> rem tecla apenas o nome do jogo  
A> rem seguido de RETURN.  
A>  
A> DATE

Nova data:

Entre com a nova data onde sex é o dia da semana (dado automaticamente pelo micro), 12=dia do mês (de 1 a 31), 06=mês do ano (de 1 a 12) e 86=ano corrente. Caso não deseje mudar a data corrente no micro, tecla simplesmente RETURN (Obs.: as datas dadas acima são apenas para exemplificar a operação).  
Recomenda-se entrar com a nova data, para que os novos programas e/ou modificações nos antigos sejam corretamente datados.

Agora o sistema está pronto para receber os seus comandos. Caso nenhum dos drives acenda o led de seleção ou os dois acendam ao mesmo tempo, quando o computador for ligado, algum erro de conexão ou de alimentação deve estar ocorrendo.

A letra que precede o sinal ">" indica o drive que receberá os comandos digitados após pressionar a tecla RETURN. Para mudar esta condição, digite a letra do drive desejado, seguida de "." e < RETURN >

EX:

A > (drive A como titular)  
A > B: < RETURN > (comando de mudança de drive titular)  
B > (drive B como titular)

Os nomes dos arquivos podem ter de 1 a 8 caracteres e suas extensões, de 1 a 3. Podem ser digitados em letras maiúsculas ou minúsculas, separadas por um ponto, sendo registrados pelo DDX-DOS em letras maiúsculas. Quando selecionado, o nome do arquivo pode vir acompanhado da designação do drive em que reside. Quando este for o drive titular, o acompanhamento da designação é desnecessário.

Então, o formato para trabalhar com o arquivo é o seguinte:

(designação do drive):(nome do arq).(extensão)

Os seguintes caracteres podem ser utilizados na constituição dos nomes dos arquivos e suas extensões:

de A a Z

de 0 a 9

\$ & # % ' ( ) - @ \ ^ \_ { } | !

### 2.1.1 - CARACTERES ESPECIAIS

Dois caracteres especiais (wild cards) podem ser usados para simplificar o seu trabalho com o nome dos arquivos e suas extensões: O asterisco (\*) e o sinal de interrogação (?).

#### 2.1.1.A - Caracter interrogação (?)

O sinal de interrogação em um arquivo ou extensão indica que qualquer caracter pode ocupar aquela posição.

Por exemplo, o comando:

DIR AUTO? XEC. BAT

#### 2.1.1.B - CARACTER ASTERISCO (\*)

Representando tanto um caracter indefinido quanto um grupo deles, o asterisco pode ser utilizado como parte de um nome, junto a caracteres definidos, ou substituí-lo por inteiro, tomando-o um nome indefinido. Usa-se no nome do arquivo ou em sua extensão.

EX: DIR AUTO \*.BAT

Este comando lista todo arquivo cujo nome comece com AUTO e tenha uma extensão.BAT  
DIR \*.BAT

Lista todo arquivo com a extensão \*.BAT.

A configuração de caracteres especiais se refere a todos os arquivos do drive. Note que esta configuração pode ser muito poderosa e destrutiva, quando usadas nos comandos do DDX-DOS.

Por exemplo: o comando DEL \*.\* apaga todos os arquivos do drive, independente do nome do arquivo ou extensão.

### 2.1.2 - NOMES DE ARQUIVOS ILEGAIS.

Alguns arquivos são tratados particularmente pelo DDX-DOS, sendo designados nomes de 3 letras para selecioná-los. Estes nomes relacionados abaixo, não podem ser usados para designar arquivos ou extensões.

**AUX** – Refere-se a uma entrada de ou saída para algum mecanismo auxiliar (tanto uma impressora como um drive).

**CUN** – Usado para referir-se a uma entrada do teclado ou saída para o console do terminal (vídeo).

**NUL** – Usado quando não se quer criar um arquivo particularmente, mas o comando requer um arquivo de entrada ou saída.

**LST** ou **PRN** – Para referir-se à impressora.

Se for utilizado qualquer um destes nomes como um arquivo ou extensão, será relacionado aos mecanismos acima citados.

Por exemplo: **CON.XXX** refere-se ao console e não ao nome de um arquivo.

### 2.1.3. DIRETÓRIOS

O diretório contém informação sobre o tamanho dos arquivos, a sua localização no disco e a data em que foram criados.

### 2.1.4 TIPOS DE COMANDOS DDX-DOS

Existem dois tipos de comandos **DDX-DOS**:

Comandos Internos

Comandos Externos

Comandos internos são os mais simples e geralmente mais usados. Estes não são vistos, quando se faz um diretório do disco; fazem parte do processador de comandos. Quando digitados, são executados imediatamente.

Consistem em:

<b>BASIC</b>	<b>DIR</b>	<b>REN</b>
<b>COPY</b>	<b>FORMAT</b>	<b>REN (RENOMEAR)</b>
<b>DATE</b>	<b>MODE</b>	<b>TIME</b>
<b>DEL (ERASE)</b>	<b>PAUSE</b>	<b>TYPE</b>
		<b>VERIFY</b>

Comandos externos residem em disco como arquivos e devem ser lidos, ou carregados do disco, para depois serem executados. Se o disco, que contém o comando, não estiver no drive, o **DDX-DOS** não conseguirá encontrá-lo nem executá-lo.

Qualquer arquivo com extensão **.COM** ou **.BAT** é considerado um comando externo. Por exemplo, **FILCON.COM** e **COMP.COM**. Pelo fato de todos os comandos externos residirem em disco, pode-se criar comandos e adicioná-los ao sistema. Programas que forem criados com vários tipos de linguagem (incluindo **Assembler**) serão **.COM** (executáveis) comandos.

A extensão não necessita ser digitada para entrada destes. Basta o nome seguido de **RETURN**.

### 2.1.5. OPÇÕES EM COMANDOS

Opções podem ser incluídas nos comandos **DDX-DOS** para especificar informação adicional ao sistema. Se não forem incluídas algumas opções, o **DDX-DOS** assumirá um valor padrão.

O formato de todos os comandos é o seguinte:

**COMANDO (OPÇÕES)...**

Tipos de opções:

**Interruptores:** são opções que controlam os comandos **DDX-DOS**. Eles são precedidos por uma barra (ex. /p).

**Argumentos:** São opções que fornecem maior informação aos comandos. Geralmente são utilizados para delimitar processos, por exemplo **ON** ou **OFF**.

**Especificação de Arquivo:** Refere-se a designação do drive e de um arquivo com sua extensão opcional de 3 letras do seguinte formato:

< d : > < nome do arquivo > < .ext >

d : Refere-se a designação do drive

**Arquivo:** Refere-se a qualquer nome válido para um arquivo de disco, incluindo uma extensão opcional. Não deve referir-se a nenhum mecanismo nem à designação de drive.

**Ext.:** refere-se a uma extensão opcional consistindo em um ponto e de 1 a 3 caracteres. Quando usada, a extensão virá após o nome do arquivo.



## 2.1.6 INFORMAÇÕES COMUNS A TODOS OS COMANDOS DDX-DOS

- Comandos são geralmente seguidos de uma ou mais opções.
- Comandos e opções podem ser digitados em maiúsculas, minúsculas ou uma combinação de ambas.
- Comandos e opções devem ser separadas por delimitadores. Estes geralmente são um espaço ou uma vírgula.

Ex: DEL ARQUIVO.VEL NOVOAR. TXT  
RENAME, ESTEARQ AQUELEAR.

Podem ser utilizados também como delimitadores nos comandos DDX-DOS o ponto e vírgula (;), o sinal de igualdade (=) ou a tecla TAB.

- Não separe um nome de arquivo com delimitadores: tanto dois pontos como ponto final também são delimitadores.

• Quando as instruções dizem "Aperte qualquer tecla quando pronto" pode-se apertar qualquer tecla, menos <CONTROL-C>

- É necessário incluir a extensão, quando se referir a um arquivo que já possua uma.

Um comando pode ser abortado enquanto ele está sendo executado, pressionando <CONTROL-C>

Comandos somente são executados após pressionar a tecla (RETURN).

Caracteres especiais e nomes de mecanismos, por ex: PRN OU CON, não devem ser incluídos no nome de nenhum comando.

• Quando comandos geram uma grande quantidade de informação mandadas para tela, esta automaticamente rodará a imagem para a entrada da próxima linha. Para dar uma pausa neste processo, que impedirá que sejam lidas todas as informações na tela, pressione <CONTROL-S> (PAUSA).

• As chaves de função e edição podem ser usadas quando se entra com comandos. Leia o item 2.1.11, para uma descrição completa destas chaves. A letra que precede o sinal " > " indica o drive titular.

- O acionador de disco flexível é um drive fonte ou um drive destino.

Este é para onde a informação é transferida e aquele é de onde a informação é retirada.

## 2.1.7 PROCESSANDO BATERIAS (BATCH)

Com o DDX-DOS, pode-se colocar uma sequência de comandos em um arquivo especial chamado bateria. "Bateria" de comandos são processados assim como foram digitados no terminal. Todo arquivo deve ter a extensão .BAT e é processado digitando o seu nome, sem a extensão.

Dois comandos DDX-DOS são válidos para o uso específico nos arquivos bateria, REME PAUSE. REM permite que se faça comentários e marcações em um BATCH sem serem interpretados como comandos. PAUSE mostra uma mensagem operacional e permite continuar ou abortar a execução do BATCH em um determinado ponto.

A lista a seguir contém informações que devem ser lidas, antes de elaborar um BATCH com o DDX-DOS:

- Não entre com o nome "BATCH" (a não ser que o arquivo a processar-se chame BATCH.BAT).
- Somente o nome do arquivo deve ser digitado para a execução da bateria. Não digite a extensão.
- Se <CONTROL-C> for digitado durante uma bateria, aparecerá a seguinte mensagem:

TERMINA BATCH (S/N)?

Se S for digitado, o restante dos comandos existentes na bateria serão ignorados e o sinal do sistema (x) aparecerá imediatamente. Se N for digitado, somente o comando corrente terminará e o resto dos comandos serão processados normalmente.

• Caso o disco, que contém o arquivo bateria que está sendo executado, for retirado do drive, o DDX-DOS avisa que o disco deve ser reinserido para que o próximo comando possa ser carregado.

- O último comando de uma bateria pode ser o nome de outra.

Isto permite que se chame uma .BAT a partir de outra, quando a primeira termina.

## 2.1.8 O ARQUIVO AUTOEXEC.BAT

Quando se liga o micro, carregando o DDX-DOS, o processador procura um arquivo chamado AUTOEXEC.BAT. Este arquivo é automaticamente carregado e executado, quando encontrado no disco. Lembre-se que AUTOEXEC.BAT indica uma bateria de comandos.

Se o DDX-DOS encontrar este arquivo, a entrada da data é omitida.

## 2.1.9 COMO CRIAR UM ARQUIVO BATERIA (BATCH).

Se, por exemplo, for desejado carregar automaticamente o BASIC e rodar um programa chamado MENU ao iniciar o DDX-DOS, pode-se criar um arquivo BAT da seguinte forma:

1 - Digite:

```
COPY CON: AUTOEXEC.BAT
```

Este comando aciona o DDX-DOS para copiar a informação do teclado no AUTOEXEC. BAT. Note que este arquivo necessita ser criado no diretório do disco.

2 Digite:

```
BASIC MENU
```

Este comando está no AUTOEXEC.BAT. Ele faz o sistema operacional acionar o BASIC, carregar e rodar o programa MENU.

3. Tecle <CONTROL-Z> e <RETURN> para colocar o comando BASIC MENU no arquivo AUTOEXEC.BAT.

O programa MENU rodará agora automaticamente sempre que se inicializar o DDX-DOS.

Para rodar um outro programa BASIC, coloque o nome do programa do MENU no item 2 do exemplo anterior. Pode-se entrar com qualquer comando DDX-DOS ou uma série deles nos arquivos bateria.

A entrada da data será omitida pelo DDX-DOS a menos que você a coloque como parte da bateria.

Recomenda-se que seja feito assim, para que o disco possua um diretório corretamente datado.

## 2.1.10 PARÂMETROS VARIÁVEIS NOS ARQUIVOS. BAT

Talvez haja necessidade de se criar programas aplicativos e rodá-los com diferentes grupos de dados. Estes dados podem ser armazenados em vários arquivos.

Quando usado nos comandos DDX-DOS, um parâmetro é uma opção que o operador define. Também podem ser criados arquivos (BAT) com parâmetros variáveis. Estes parâmetros de %0 a %9, podem ser substituídos por valores durante a execução de arquivo (BAT).

Por exemplo, quando se digita o comando COPY CON MEUARQ.BAT, as próximas linhas que digitar serão copiadas no arquivo chamado MEUARQ, no drive titular:

```
A > COPY CON MEUARQ.BAT
COPY %1. MAC %2. MAC
TYPE %2. PRN
TYPE %0. BAT
```

Agora tecla (CONTROL-Z) e (RETURN). O DDX-DOS fornece na tela a seguinte mensagem:

1 Arquivo copiado

A >

O arquivo MEUARQ.BAT, que consiste em três comandos, reside agora no drive titular.

Os parâmetros variáveis %1 e %2 são substituídos sequencialmente pelos valores que forem fornecidos ao executar o arquivo. O parâmetro variável %0 é substituído pela designação de drive, se especificada, e o nome do arquivo bateria (por exemplo MEUARQ).

OBS.:

1. Podem ser especificados até 10 parâmetros (%0 a %9)

2. Se for usado ainda o sinal de porcentagem como parte do nome do arquivo é necessário digitá-lo duas vezes.

Por exemplo, para especificar o arquivo ABC%. COM, deve ser digitado ABC%%.COM no arquivo (.BAT).

Para executar o arquivo MEUARQ.BAT e especificar os parâmetros variáveis, deve-se entrar com o nome do arquivo (sem a extensão), seguido pelos parâmetros que devem substituir %1, %2 etc.

Lembre-se que MEUARQ.BAT consiste em 3 linhas:

```
COPY %1.MAC %2.MAC
TYPE %2.PRN
TYPE %0. BAT.
```

Para executar o processamento do MEUARQ, digite:

```
MEUARQ A: PROG1 B: PROG2
```

MEUARQ é substituído por %0, A: PROG1 por %1 e B: PROG2 por %2.

O resultado é o mesmo que se tivesse digitado todos os comandos de MEUARQ com os seus parâmetros, como a seguir:

```
COPY A:PROG1.MAC B:PROG2.MAC
TYPE B: PROG2.PRN
TYPE MEUARQ.BAT
```

A tabela a seguir ilustra como o DDX-DOS faz a substituição dos parâmetros:

NOME	PARÂMETRO1(%0) (MEUARQ)	PARÂMETRO2(%1) (PROG1)	PARÂMETRO3(%2) (PROG2)
MEUARQ	MEUARQ.BAT	PROG1.MAC	PROG2.MAC PROG2.PRN

Lembre-se que o parâmetro variável %0 sempre é substituído pela designação do drive (se especificado), e o nome do arquivo.BAT.

## 2.1.11 TECLAS DE EDIÇÃO E CONTROLE DO DDX-DOS.

Teclas de Edição Especial do DDX-DOS  
Funções dos Caracteres de Controle

### 2.1.11A - TECLAS DE EDIÇÃO ESPECIAIS DO DDX-DOS

As chaves de edição especiais necessitam de uma ênfase, porque a maioria dos sistemas operacionais as utilizam para manusear a entrada de comandos. Não é necessário digitar a mesma sequência de teclas repetidamente, porque a última linha de comando é automaticamente armazenado em uma área de memória especial.

Usando esta área e as chaves de edição, pode-se tirar vantagem dos seguintes recursos do MSX:

- Uma linha de comando pode ser instantaneamente repetida operando duas chaves.
- Se um erro for cometido na linha de comando, é possível editá-la e consertá-la sem ter que digitar tudo novamente.
- Uma linha de comando, que é similar à anterior, pode ser editada e executada com um mínimo de digitação usando as chaves de edição especiais.

A relação entre a linha de comando e a área de memória está ilustrada abaixo.

ENTRADA DO USUÁRIO

↓  
LINHA DE COMANDO

↓  
COMMAND.COM



ÁREA DE MEMÓRIA

CHAVES DE EDIÇÃO ESPECIAIS

NOME	CHAVE	FUNÇÃO
COPY1	Flecha direita ^ Y	Copia um caracter da linha anterior para a nova linha
Copy up	Select ^ X	Copia todos os caracteres da linha anterior até o caracter especificado
COPYALL	Flecha p/baixo ^ _	Copia todos os caracteres da linha anterior para a nova linha
SKIP1 SALTAR1	DEL	Salta um caracter da linha anterior
SKIPUP SALTA ATÉ	CLS ^ L	Salta os caracteres da linha anterior até o caracter especificado
VOID NULO	Flecha p/cima ESC ^ A ^ U ^ C	Anula a linha corrente, deixa a área de memória intacta

BS	Flecha p/esquerda BS ^H ^L	Apagar o último caracter digitado
INSERT	INS ^R	Entrada/Saída, modo de inserção
NEWLINE NOVALINHA	Home ^K	Transforma a linha corrente na nova área de memória.

#### EXEMPLO:

Se o comando `DIR PROG.COM` for digitado,

O `DDX-DOS` mostrara informações sobre o arquivo:

`PROG.COM` na tela. A linha de comando estará guardada na área de memória. Para repetir o comando, digite apenas duas teclas: `(COPYALL)` e `(RETURN)`

O comando repetido será mostrado na tela da seguinte forma

```
< COPYALL > DIR PROG.COM < RETURN >
```

Note que, teclando `(COPYALL)`, o conteúdo da área de memória especial é transferido para a linha de comando; teclando `(RETURN)`, a linha de comando passa para o processador de comandos para ser executada.

Se for preciso obter informações sobre um arquivo chamado `PROG.ASM`, pode ser usada a área da memória e teclar:

```
< COPYUP > C.
```

Teclando `< COPYUP >`, todos os caracteres da área de memória até o "C", sem incluí-lo, são copiados, ficando a linha de comando como a seguir:

```
DIR PROG
```

Note que o traço depois do ponto é o cursor. Agora digite:

● `ASM`

O resultado é:

```
DIR PROG.ASM
```

A linha de comando está agora na área de memória e pronta para ser entregue ao processador de comandos para execução. Para fazer isto, teclando `< RETURN >`

Suponha que se queira executar o seguinte comando:

```
TYPE PROG.ASM
```

Para isto teclando:

```
TYPE < INSERT > < COPYALL > < RETURN >
```

Note que enquanto se tecla, os caracteres entram diretamente na linha de comando e sobrepõem-se aos caracteres correspondentes na área de memo. Esta reposição automática é inibida quando a tecla `INSERT` é pressionada. Então, os caracteres "TYPE" substituem "DIR" na área de memo. Para inserir um espaço entre "TYPE" e "PROG.ASM", pressione `INSERT` e, logo após, a barra de espaço. Finalmente, para copiar o resto da área de memo para a linha de comando, pressione `< COPYALL >` e `< RETURN >`. O comando "TYPE PROG.ASM" é processado e carregado na área de memo.

Caso se digite "BYTE" em vez de "TYPE", um erro de comando ocorrerá.

Esta linha mal digitada ainda pode ser salva, antes do `< RETURN >` ter sido teclado, criando uma nova área de memo com a chave (`NEWLINE`):

```
BYTE PROG.ASM < NEWLINE >
```

Este comando errôneo pode ser reeditado, digitando:

```
T < COPY 1 > P < COPYALL >
```

A chave (`COPY1`) tranlere um único caracter da área de memo para a linha de comando. A linha resultante será o comando desejado:

```
TYPE PROG.ASM
```

Essas chaves de edição especiais podem facilitar o trabalho no teclado. A próxima seção descreve as funções dos caracteres de controle que também podem ajudar na digitação dos comandos.

## 2.1.11 B FUNÇÕES DOS CARACTERES DE CONTROLE

Consistem em funções que afetam a linha de comando.  
Lembre-se que para ser digitado um caracter de controle, como CONTROL, deve-se manter a tecla de controle pressionada e pressionar a tecla "C".

TABELA DE FUNÇÕES DOS CARACTERES DE CONTROLE

CARACTER	FUNÇÃO
(CONTROL-N)	Cancela o processamento de informações para impressora.
(CONTROL-C)	Aborta o comando corrente.
(CONTROL-H)	Remove o último caracter do comando e o apaga no terminal de vídeo.
(CONTROL-J)	Faz um fim de linha física, mas não esvazia a linha de comando. Use a tecla (LINE FEED) para estender a corrente linha além do limite físico de um terminal de vídeo.
(CONTROL-P)	Transfere a informação do vídeo para impressora.
(CONTROL-S)	Suspende a saída de informação para o terminal de vídeo. Aperte qualquer tecla para continuar.

## 2.1.12 - INSTRUÇÕES PARA USUÁRIOS DE SISTEMAS COM UM ÚNICO DRIVE.

Num sistema com um único drive, os comandos são como se fossem num sistema multi-drive. A e B representam dois discos.

Se o drive B foi especificado, quando o "disco do drive A" for o último a ser usado, é necessário inserir o disco para o drive B. Por exemplo:

```
A > COPY COMMAND. B:  
Insira disquete como drive B:  
Aperte qualquer tecla quando pronto.  
1 arquivo copiado.  
A > _
```

Se o drive A for especificado, quando o "disco do Drive B" for o último a ser usado, o disco deve ser novamente trocado. Neste caso, o DDX-DOS solicita a inserção do "disco do Drive A".

O mesmo procedimento é seguido, quando um comando é executado de um arquivo bateria (BATCH). O DDX-DOS espera o disco apropriado ser inserido e qualquer tecla ser pressionada para prosseguir.

OBS.: A letra que aparece antes do sinal ">" representa o drive titular onde o DDS-DOX procurará o arquivo cujo nome for digitado sem especificação de drive. Esta letra não representa o último drive utilizado. Por exemplo, assumo que A é o drive titular. Se a última operação efetuada foi DIR B:, o DDX-DOS acredita que o "disco do drive B", continua no drive. Não obstante, o sinal continua A, porque o A continua sendo o drive titular. Ao digitar DIR, o DDX-DOS solicitará a inserção do "disco do drive A", porque este é o titular e no comando DIR não foi especificado nenhum outro.

## 2.1.13 - ERROS DE DISCO.

Se um erro de disco ocorre durante a execução de um comando ou programa, o DDX-DOS tenta a operação três vezes. Se a operação não é lograda, uma das mensagens seguintes é apresentada:

### DISCO PROTEGIDO

Esta mensagem indica que o disco, que está inserido, possui uma proteção contra a gravação.

### NÃO PRONTO

Esta mensagem indica que o drive selecionado não está pronto para a operação.

### ERRO DE DISCO - ??

Aborta, ignora, Tenta de novo

O DDX-DOS aguarda a entrada de uma das alternativas:

A - Aborta, termina a operação voltando ao sistema.

I - Continua a operação ignorando o erro.

T - Tenta de novo. Repete a operação. Esta opção será utilizada quando o operador tenha corrigido o erro.

## 2.2. GUIA DOS COMANDOS DDX-DOS.

OBS.: Usuários de sistema com um único drive devem consultar o item 2.1.12 para os procedimentos adicionais requeridos no manuseio de alguns dos seguintes comandos.

Os comandos DDX-DOS serão descritos a seguir. Note que sinônimos dos mesmos estão entre parênteses.

BASIC	Vai para o DDX-BASIC
COPY	Copia arquivo(s) especificado(s)
DATE	Apresenta e corrige a data.
DEL	Apaga arquivo(s) especificado(s) (ERASE)
DIR	Lista de arquivos existentes.
FORMAT	Formata o disco para receber um arquivo DDX-DOS
MODE	Seleciona o modo do terminal de vídeo.
PAUSE	Dá uma pausa em um arquivo bateria.
REM	Mostra um comentário em um arquivo bateria.
REN	Renomeia o primeiro arquivo como segundo arquivo (renomeia).
TIME	Apresenta e corrige a hora (se o computador possuir relógio interno).
TYPE	Apresenta o conteúdo de um arquivo especificado.
VERIFY	Liga/Desliga o modo de verificação.

### BASIC

Sintaxe: BASIC < Espec. arquivo >

PROPÓSITO: Inicializar o DDX-BASIC

COMENTÁRIOS: Este comando inicializa o Basic de disco do DDX-DOS. Se o arquivo Basic for especificado pelo < Espec. arquivo >, o programa será automaticamente carregado e rodado após a inicialização do Basic.

Este comando muda o slot para efetivar a ROMBASIC.

O "mapeamento da memória" é diferente entre o DDX-DOS e o basic de disco DDX.

Use "CALL SYSTEM" para voltar ao sistema DDX-dos.

### COPY

SINTAXE: COPY < Arq. espec > < arquespec >

PROPÓSITO: Copia um ou mais arquivos em outro disco.

Se houver preferência, nomes diferentes poderão ser dadas às cópias. Este comando pode também copiar arquivos no mesmo disco de origem.

COMENTÁRIOS: Se o segundo < ARQESPEC > não for fornecido, a cópia será no drive titular e terá o mesmo nome que o arquivo original. Se o primeiro < ARQ ESPEC > está no drive titular e o segundo ARQESPEC não for especificado, o comando COPY será abortado. (Arquivos não podem ser copiados neles mesmos).

O DDX-DOS apresentará a seguinte mensagem de erro:

Arquivo não pode ser copiado nele mesmo

O(s) arquivo(s) copiado(s).

A segunda opção < ARQESPEC > pode ser de três formas:

1. Se for só uma designação de drive (d:), o arquivo original será copiado com o mesmo nome no drive especificado.

2. Se for só o nome do arquivo, o arquivo original será copiado em um arquivo no drive titular com nome especificado.

3. Se for uma especificação completa, o arquivo original será copiado no drive com o nome especificado.

O comando COPY permite também a soma de arquivos durante a cópia. O comando é constituído pela simples enumeração de arquivos como se fossem opções de COPY, separadas por "+".

Por exemplo:

COPY A.XYZ+B.COM + B: C.TXT GDEARQ.CRP

Este comando soma os arquivos chamados: A.XYZ, B.COM e B:C.TXT, e coloca-os em um arquivo do drive titular chamado GDEARQ.CRP.

Para combinar vários arquivos, usando caracteres especiais, pode ser digitado:

COPY\*.LST COMBIN.PRN

Este comando pega todos os arquivos com a extensão .LST e combina-os em um arquivo chamado COMBIN.PRN.

No seguinte exemplo, cada arquivo \*.LST encontrado será combinado com o arquivo \*.REF correspondente (de mesmo nome). O resultado será armazenado em um arquivo com mesmo nome mas com extensão .PRN. Portanto, ARQL.LST será combinado com ARQL.REF para formar ARQL.PRN e assim por diante.

## COPY \*.LST + \*.REF \*.PRN

Não entre com um COPY somador no qual um dos arquivos fonte possui a mesma extensão que o destinatário.

Por exemplo, o seguinte comando é um erro, se o arquivo TODOS.LST já existe:

```
COPY *.LST TODOS.LST
```

O erro não será detectado, porém, até que TODOS.LST seja adicionado. Até então, ele já poderá ter sido destruído.

COPY compara o nome do arquivo fonte com o nome do destinatário. Se estes forem iguais, um arquivo de entrada é saltado e a mensagem de erro "conteúdo do destino perdido após a cópia" aparece na tela. A seguir, há um exemplo importante:

```
COPY TODOS.LST+*.LST
```

Este comando anexa todos os arquivos \*.LST, exceto TODOS.LST, ao arquivo TODOS.LST. Não produz nenhuma mensagem de erro e é a forma correta para somar arquivos usando o comando COPY.

Pelos arquivos em ASCII serem usualmente somados, este comando interpreta um CTRL+Z(1AH) como um "fim de arquivo" escrito no arquivo.

Existe necessidade de uma terminação "/B" para usar um fim de arquivo físico, quando arquivos binários precisam ser somados.

```
COPY/B A.COM+B.COM
```

Neste exemplo, o B.COM é somado após o A.COM e o arquivo destino continua sendo A.COM.

Qualquer arquivo pode ser somado usando "/B" para arquivos binários e "/A" para arquivos em ASCII. A terminação é válida para o arquivo em que está e até outra terminação aparecer.

Se um CTRL-Z for ou não adicionado ao final do arquivo destino, estará implícito por uma terminação do mesmo. Nenhum CTRL-Z no arquivo fonte é interpretado como "/A". Apenas um CTRL-Z é colocado, quando um arquivo é escrito em eleito de "/A". Entretanto, mais CTRL-Z são adicionados, como a seguir:

```
COPY A.ASM/B B.ASM/A
```

Neste exemplo, "/B" dispensa a remoção de CTRL-Z e "/A" adiciona um CTRL-Z. Quando não há soma, as terminações "/A" e "/B" são válidas e o tipo de arquivo padrão é binário. A terminação "/A" cancela a cópia no primeiro CTRL-Z.

## DATE

SINTAXE: DATE <dd> - <mm> - <aa>

PROPÓSITO: Mudar a data vigente ou entrar com nova data no sistema. Esta data será gravada no diretório de qualquer arquivo que for criado ou alterado. A data pode ser mudada digitando o comando ou um arquivo bateria (BATCH). O DDX-DOS não pede a data automaticamente se for inicializado com um arquivo. BAT (recomenda-se incluir este comando no arquivo bateria para ter o diretório sempre corretamente datado).

COMENTÁRIOS: Se DATE for digitado, este comando responderá com a seguinte mensagem:

Data atual e (dia) (dd) - (mm) - (aa)

Nova data:

Pressione RETURN se a data estiver correta.

É possível também teclar uma data em particular, após o comando DATE:

```
DATE 8-10-85
```

Neste caso, não será necessário a apresentação da antiga data com a solicitação de "nova data".

A nova data deve constituir-se apenas de números.

As opções são:

(dd) = 1-31

(mm) = 1-12

(aa) = 0-79, 80-99 ou 1980-2099

O dia, mês e ano devem ser separados por hífens (-) barras (/) ou pontos (.). O DDX-DOS está programado para alterar meses e anos corretamente, quando o mês tiver 31, 30, 29 ou 28 dias. Trabalha também com anos bissextos.

O (aa) é um número de dois dígitos de 80 a 99 (o 1900 é assumido), ou de 00 a 79 (o 2000 é assumido) ou também um número de quatro dígitos de 1980 a 2099.

Se as opções ou as separações não são válidas, a seguinte mensagem aparecerá:

Data inválida

Nova data: \_ e o comando espera a entrada de uma nova data.

OBS.: A ordem da data (dia-mês-ano) pode mudar dependendo da versão. Por exemplo, temos "mês-dia-ano" na versão americana e "ano-mês-dia" na versão japonesa.

<b>DEL</b>	
<b>SINÔNIMO:</b>	DELETE,ERASE
<b>SINTAXE:</b>	DEL<arq>espc >
<b>PROPÓSITO:</b>	Apaga todos os arquivos especificados.
<b>COMENTÁRIOS:</b>	Se o <arq>espc> é *, a mensagem "tem certeza" aparecerá. Caso um "S" ou RETURN seja teclado como resposta, então todos os arquivos residentes no drive selecionado serão deletados, apagados. ERASE ou DELETE também podem ser digitados.

### DIR

**SINTAXE:** DIR <arq>espc > (/P) (/W)

**PROPÓSITO:** Listar arquivos do diretório do drive selecionado.

**COMENTÁRIOS** - Se for digitado somente DIR, todos os arquivos do drive titular serão listados. Quando só a especificação de drive for dada (DIR d:) todos os arquivos do drive especificados serão listados. Caso só um nome de arquivo for especificado sem extensão, então todos os arquivos com o nome especificado no drive titular serão listados. Se for designado um arquivo (por exemplo, DIR d: arq. ext), todos os arquivos designados no disco do drive designado serão listados. Em todos os casos, os arquivos são listados com a sua dimensão em bytes, com a data em que foi criado ou com a sua última modificação e a hora em que ocorreu.

Os caracteres especiais? e\* (sinal de interrogação e asterisco) podem ser usados na opção de arquivos. Para sua conveniência, os seguintes comandos DIR são equivalentes:

<b>COMANDO</b>	<b>EQUIVALENTE</b>
DIR	DIR *.*
DIR Arquivo	Dir Arquivo.*
DIR .EXT	DIR *.EXT
DIR .	DIR *

Duas terminações podem ser especificadas com DIR.

A terminação /P seleciona o modo de paginação. Com /P, a listagem do diretório dá uma pausa ao preencher toda a tela. Para continuar a mostragem, pressione qualquer tecla.

A terminação /W faz com que os arquivos sejam listados sem nenhuma especificação e quantos nomes couberem por linha.

### FORMAT

**Sintaxe:** FORMAT

**PROPÓSITO:** Formata o disco no drive especificado para aceitar os arquivos DDX-DOS.

**COMENTÁRIOS** - Este comando inicializa o diretório e divide o disco para receber os arquivos. Um disco novo deve ser formatado antes do uso. Se um disco usado for formatado, todos os arquivos existentes nele serão destruídos.

O DDX-DOS mostra a seguinte mensagem:

Qual drive? (A, B)

Selecione o nome do drive com cuidado; logo aparecem as opções:

- 1 - 40 trilhas face simples
- 2 - 40 trilhas face dupla
- 3 - 80 trilhas face simples
- 4 - 80 trilhas face simples

Escolha de acordo com a configuração do sistema utilizado; após feita a escolha, aparecerá na tela:

Tecla algo quando pronto

Insira o disco a ser formatado no drive selecionado e pressione qualquer tecla para dar início à formatação.

Quando a formatação termina, o DDX-DOS dará a seguinte mensagem:

Formatado



## MODE

**SINTAXE:** MODE < largura >

**PROPÓSITO:** largura é o número máximo de caracteres por linha do vídeo. Deve ser entre 1 e 40 se for 32 ou menos, o modo de tela selecionado é 1, mais que 32, o modo 0 é selecionado.

O modo de tela padrão e larguras das versões internacionais são descritos a seguir:

Versão	Modo de tela padrão	Largura de tela padrão
JAPÃO	1	29
USA		39
INGLATERRA		
DINAMARCA	0	37
FRANÇA		
INT		

## PAUSE

**SINTAXE:** PAUSE [comentário]

**PROPÓSITO:** Durante a execução de um arquivo bateria, pode ser preciso muda discos ou efetuar alguma outra operação. PAUSE suspende a execução até que qualquer tecla seja pressionada, exceto <CONTROL-C>.

Quando o processador de comandos encontra um PAUSE, ele mostra a seguinte mensagem:

Aperte qualquer tecla...

Se for pressionado <CONTROL-C>, outra mensagem aparecerá:

Termina BATCH (S/N)?

Se um "S" for teclado como resposta, a execução do comando corrente da bateria será abortado e o controle voltará ao sistema operacional. PAUSE também pode ser usado para separar um BACH em partes, permitindo ao operador finalizá-lo em um ponto intermediário.

O comentário é opcional e entra na mesma linha do PAUSE. Pode-se querer transmitir ao usuário alguma mensagem durante a pausa. Por exemplo, avisá-lo que deve mudar o disco de algum dos drives. O comentário aparecerá antecedendo o "Aperte qualquer tecla...".

## REM

**SINTAXE:** REM [comentário]

**PROPÓSITO:** Apresentar mensagens que estão na mesma linha que o comando REM em um BATCH durante a execução deste.

**COMENTÁRIOS:** As únicas separações permitidas no comentário são espaço, TAB e vírgula.

## REN

**SINÔNIMO:** RENAME

**SINTAXE:** REN < arqespec > < arqespec >

**COMENTÁRIOS** - A primeira opção deve ser acompanhada da designação do drive, se ela não estiver no drive titular. Qualquer designação do drive na segunda opção é ignorada.

O arquivo permanecerá no disco de origem.

Os caracteres especiais (?, \*) podem também ser usados neste comando. Todos os arquivos correspondentes ao primeiro arquivo são renomeados. Se os caracteres especiais aparecerem no segundo arquivo, os caracteres correspondentes serão alterados.

Por exemplo, o comando a seguir muda o nome de todos os arquivos com a extensão. LST para nomes similares com a extensão. PRN:

REN \*.LST \*.PRN

No próximo exemplo, REN renomeia o arquivo CIPOA do drive B: para COPIA:

REN B:CIPOA ?0?1?

O arquivo permanece no drive B.

Uma tentativa de renomear um arquivo para um nome já existente no disco resultará em uma mensagem de erro: "ERRO NO RENOME".

## TIME

**SINTAXE:** TIME <hh> : <mm> : <ss>

**PROPÓSITO:** Mostra e muda a hora.

**COMENTÁRIOS** - Se o comando TIME for digitado sem nenhum argumento, a seguinte mensagem aparecerá:

Hora atual é <hh> : <mm> : <ss> . <cc>

Nova hora:

Pressione RETURN se não quiser mudar a hora apresentada. Uma nova hora pode ser dada com a opção para o comando TIME. EX:

TIME 8:20

A hora deve ser entrada usando somente números; letras não são permitidas. As opções são:

<hh> = 00 a 24

<mm> = 00 a 59

<ss> = 00 a 59

A hora e os minutos devem ser separados por dois pontos (:). Não é necessário digitar os segundos nem os centésimos de segundos.

O DDX-DOS usa a hora digitada como a nova hora, se as opções e separações forem válidas. Caso isto não aconteça, aparece a mensagem:

Hora inválida

Nova hora: \_

**OBS.:** Se o computador utilizado não tem relógio interno, este comando não é válido.

## TYPE

**SINTAXE:** TYPE <arq espec>

**PROPÓSITO:** Mostra o conteúdo do arquivo especificado.

**COMENTÁRIO:** Use este comando para examinar um arquivo sem modificá-lo. O único formato performado pelo TYPE é que "tabs" são expandidos para espaços consistentes com paradas de tab (tabstops) a cada oito colunas. Note que uma mostragem de arquivos binários envia para o micro caracteres de controle (como CTRL-Z), incluindo sons, "form feeds" e rotinas de escape.

## VERIFY

**SINTAXE:** VERIFY (ON/OFF)

**PROPÓSITO:** Ligar ou desligar o modo de verificação (ler após gravar).

**COMENTÁRIOS:** O comando VERIFY ON aciona o modo de verificação. Sempre que alguns dados são gravados no disco, esses dados são lidos do disco e verificados. Se a verificação não estiver correta, ocorrerá um "ERRO DE E/S". O comando VERIFY OFF desliga o modo de verificação. O modo padrão é VERIFY OFF. Utilizando este comando, a gravação é mais confiável, mas precisa de um tempo maior.

### 2.3 Guia de Basic de Disco MSX

#### 2.3.1. Comandos e mostragens

BLOAD

BSAVE

CLOSE

COPY

DSKO

FIELD

FILES e LFILES

FORMAT

GET

INPUT #

KILL

LINE INPUT #

LOAD  
LSET E RSET  
MAXFILES  
MERGE  
NAME  
OPEN  
PRINT # e PRINT # USING  
PUT  
RUN  
SAVE  
SYSTEM  
VERIFY

#### BLOAD

SINTAXE: BLOAD "<arquespec>" [(,R)|(,S)] (,offset)

PROPÓSITO: Carregar um programa em linguagem de máquina do disco ou cassete para a memória.

COMENTÁRIOS: O nome do arquivo pode ser omitido só em caso de arquivo de cassete; em disco ele deve ser identificado.

Se nenhum (off set) for especificado, o programa será carregado do endereço designado pelo comando BSAVE. Se um (off set) for especificado, o programa será carregado do endereço gravado adicionado do (off set).

A opção R, automaticamente, faz rodar o programa, após ser carregado.

A opção S carrega a imagem do vídeo guardada na RAM de vídeo pelo BSAVE, S".

Se não houver especificação de drive, o programa do drive titular será carregado.

Veja também "BSAVE"

Ex. BLOAD "MIN3"

Carrega o arquivo MIN3 na memória.

#### BSAVE

SINTAXE: BSAVE "<arquespec>", <endinicial>, <endfinal> [(, <endereço executado>)|(,S)]

PROPÓSITO: Grava o programa em linguagem de máquina corrente na memória em disco ou cassete.

COMENTÁRIOS: O programa, do <endinicial> ao <endfinal> da memória, é gravado no disco ou cassete.

Se nenhum drive for especificado, o programa é gravado no drive titular.

O <endinicial> define o endereço de execução padrão.

A opção S grava o conteúdo da RAM de vídeo no arquivo.

Veja também "BLOAD".

EXEMPLO: BSAVE "TEMPO", &HC000, &HCFFF

Grava o programa corrente na memória, de &HC000 a &HCFFF, no drive titular com o nome "TEMPO".

#### CLOSE

SINTAXE: CLOSE (( # ) <numarq> ,( # ) <numarq...> )

PROPÓSITO: Termina E/S para um arquivo de disco.

COMENTÁRIOS: <numarq> é o número pelo qual o arquivo foi definido pelo OPEN. Um CLOSE sem argumentos fecha todos os arquivos abertos.

A associação entre um arquivo particular e um número de arquivo termina com a execução de um CLOSE.

O arquivo pode ser reaberto usando o mesmo número de arquivo ou um diferente. Do mesmo modo, esse número de arquivo desassociado pode ser usado para abrir qualquer outro arquivo.

O END, CLEAR e o comando NEW também "fecham" todos os arquivos de disco automaticamente, (STOP não fecha arquivos de disco).

EXEMPLO: CLOSE #1

#### COPY

SINTAXE: COPY "<arquespec>" "TO" <arquespec>

PROPÓSITO: Copia um ou mais arquivos de um disco para outro. Se houver preferência, pode dar nomes diferentes às cópias.

Este comando pode também copiar arquivos no mesmo disco.

**COMENTÁRIOS:** A segunda opção pode ser de três formas:

1. Se a segunda opção é só uma designação de drive, o arquivo original é copiado com o mesmo nome no drive designado.
2. Se a segunda opção é só um nome de arquivo, o arquivo original é copiado no drive titular com o nome especificado.
3. Se a segunda opção é uma especificação completa, o arquivo original é copiado no drive com o nome designado.

Em um sistema com um único drive, a entrada de comandos é como num sistema multidrive.

Se o drive B foi especificado, quando o "disco do drive A" foi o último a ser usado, é necessário inserir o disco para drive B.

Por exemplo:

**COPY "A: TESTE.ASC" TO "B:"**

Após o arquivo do "disco do drive A" ser carregado na memória, a seguinte mensagem aparecerá na tela:

Insira o disco como drive B:

Aperte qualquer tecla quando pronto.

Deve-se retirar o "disco A" e inserir o "disco B".

Então, digite qualquer tecla. (Exceto CONTROL-C OU CONTROL-STOP).

Se o arquivo for pequeno, a cópia estará completada. Mas, se o arquivo for grande, será necessário trocar os discos conforme as instruções, porque partes do arquivo serão carregados e gravadas uma após a outra.

Se o drive A foi especificado, quando o "disco do drive B" foi o último a ser usado, será solicitado que troque os discos novamente. Neste instante, o Basic pedirá para inserir o "disco do drive A". Veja também a Seção 2.1.12.

## DSKO

**SINTAXE:** DSKO < num-drive > , < número-setor-lógico >

**COMENTÁRIOS:** Grava o setor que está apontado pelo conteúdo da memória de (OF351H, OF352H).  
< num-drive > é 0 para o drive titular, 1 drive A, 2 drive B.  
< num-setor-lógico > é o número que se inicia em 0. Nenhuma checagem do número do setor válido é feita.

Obs.: Esta área de memória é destruída, quando qualquer estatuto de disco (FILES, OPEN, CLOSE, PRINT #, etc) for executado.

## FIELD

**SINTAXE:** FIELD ( # ) < numarq > , < carac arq > AS < variável string >

**PROPÓSITO:** Aloca espaços para variáveis em um "buffer" de arquivo aleatório.

**COMENTÁRIOS:** Antes que um GET ou um PUT possam ser executados, deve-se executar um FIELD para formatar o buffer do arquivo aleatório.

< numarq > é o número pelo qual o arquivo foi "aberto".

< caracarq > é o número de caracteres a serem alocados à < variável string >.

Por exemplo, FIELD 1,20 AS N\$, 10 AS ID\$, 40 AS ADD\$ aloca as primeiras 20 posições (bytes) no buffer de arquivo aleatório para a variável string N\$, os 10 próximos para ID\$, e os próximos 40 para ADD\$. FIELD não coloca nenhum dado neste buffer. Veja "LSET/RSET" e "GET".

O número total de bytes alocados em um FIELD não pode exceder o número total de bytes do registro que foi especificado, quando o arquivo foi aberto.

Caso esta condição não seja satisfeita, um erro de "Fim de arquivo" ocorrerá.

Qualquer número de estatutos de FIELD podem ser executados para o mesmo arquivo, respeitando a condição anteriormente descrita. Todos FIELD vigorarão no mesmo tempo em que o comando for entrado.

Obs.: Não use uma variável do FIELD em um estatuto do INPUT ou LET. Quando o nome de uma variável é mapeado (FIELDDED), direciona-se para a posição correta no buffer de arquivo aleatório. Se um INPUT ou LET é executado com essa variável, subsequentemente a localização da variável é deslocada para um espaço de STRING.

### EXEMPLO 1:

```
10 OPEN "A:LSTFONE" AS #1 LEN=35
15 FIELD #1,2 AS RECNR$, 35 AS DUMMY$
20 FIELD #1,25 AS NOMES, 10 AS NUMFONES
25 GET #1
```

```

30 TOTAL=CVI (RECBR)$
35 FOR I=2 TO TOTAL
40 GET #1,I
45 PRINT NOMES, NUMFONES$
50 NEXT I

```

Ilustra um FIELD de múltipla definição. Na linha 15, o campo de 35 caracteres é definido para o primeiro registro, para manter o número de registros no arquivo.

No próximo loop (35-50), a linha 20 define o campo para nomes individuais e números de telefones

EXEMPLO 2:

```

10 FOR LOOP%=0 TO 7
20 FIELD #1, (LOOP% * 16) AS OFFSETS, 16 AS A$(LOOP%)
30 NEXT LOOP%

```

Mostra a construção de um FIELD, usando uma carga de elementos de tamanhos equivalentes. O resultado é equivalente a seguinte linha:

FIELD #1, 16 AS A\$(0), 16 AS A\$(1), ..., 16 AS A\$(6), 16 AS A\$(7)

EXEMPLO 3:

```

10 DIM SIZE% (NUMB%): REM carga de tamanhos de FIELD
20 FOR LOOP%= 0 TO NUMB%: READ SIZE% (LOOP%): NEXT LOOP%
30 DATA 9, 10, 12, 21, 41
...
120 DIM A$( NUMB%): REM carga de variáveis mapeadas
130 OFFSET%=0
140 FOR LOOP%=0 TO NUMB%
150 FIELD # 1, OFFSET% AS OFFSET $, SIZE% (LOOP%)
AS A$( LOOP%)
160 OFFSET% = OFFSET% + SIZE% (LOOP%)
170 NEXT LOOP%

```

Cria um field da mesma maneira que o EX2. Porém, o tamanho do elemento varia com cada elemento. A linha equivalente é:

FIELD #1, SIZE% (0) AS A\$(0), SIZE% (1) AS A\$(1), ...,  
SIZE% (NUMB%) AS A\$( NUMB%)

## FILES e LFILES

SINTAXE: FILES " < arqspec > "  
LFILES " < arqspec > "

PROPÓSITO: Mostra no vídeo ou imprime na impressora a listagem dos arquivos de um disco.

COMENTÁRIOS: Os arquivos designados pelo < arqspec > são listados.

Se o arquivo designado não existir, o erro "arq. não existe" ocorrerá. Caso nenhum nome de arquivo for especificado, todos os arquivos do drive corrente serão listados.

Caracteres especiais podem ser usados tanto no nome do arquivo como na extensão.

Se o nome do drive for especificado, o arquivo daquele drive será listado. O comando LFILES lista o programa não no vídeo, mas sim na impressora.

Ex: FILES "B: \*.BAS"

## FORMAT

SINTAXE: CALL FORMAT ou \_FORMAT

PROPÓSITO: FORMATA o disco no drive especificado para que este possa aceitar arquivos DDX-DOS.

COMENTÁRIOS: Ao entrar com este comando, a seguinte opção aparece na tela:

Nome drive? (A,B)

Selecione o drive com cuidado. Após entrar com este dado, a seguinte opção aparece:

Escolha o modo:

1 - 40 trilhas face simples

2 - 40 trilhas face dupla

3 - 80 trilhas face simples

4 - 80 trilhas face dupla

Escolha de acordo com a configuração do sistema a utilizar. Feita a escolha, aparece na tela:

Tecla algo quando pronto.

Então, insira o disco a ser formatado no drive selecionado e pressione qualquer tecla para dar início à formatação. Quando esta terminar, a seguinte mensagem aparecerá na tela:

### FORMATADO

Obs.: Se um disco usado é formatado, todos os arquivos nele residentes são destruídos. Disco novos devem ser formatados antes do uso.

### GET

SINTAXE: GET (#) < numarq > (, < num registro > )

PROPÓSITO: Interpreta um registro de um arquivo de disco sequencial aleatório para um buffer aleatório.

COMENTÁRIOS: < numarq > é o número pelo qual o arquivo foi aberto.

Se o < numregistro > for omitido, o próximo registro (após o último GET) será utilizado para o buffer. O maior número possível de registro é 4.294.967.295.

#### EXEMPLO:

```
10 OPEN "EXEMPLO.DA" AS # 1
20 FIELD # 1, 2 AS A$, 10 AS B$
30 FOR I% = 1 TO 10
40 GET # 1, I%
50 PRINT CVI (A$); B$
60 NEXT
70 CLOSE # 1
80 END
```

OBS.: Após a execução de um GET, INPUT # e LINE INPUT #, pode ser executado para ler os caracteres do buffer de arquivo aleatório.

### INPUT #

SINTAXE: INPUT # < numarq > , < lista de variáveis >

PROPÓSITO: Lê os itens de dados do arquivo de disco e os transfere para variáveis do programa.

COMENTÁRIOS: < numarq > é o número atribuído ao arquivo, quando este foi aberto. < lista de variáveis > contém os nomes das variáveis que serão designadas para os itens no arquivo. (o tipo de variável deve corresponder com o tipo designado pelo nome da variável).

Com INPUT #, nenhum sinal de interrogação é mostrado como no comando INPUT.

Os itens de dados do arquivo devem aparecer como se fossem teclados em resposta a um INPUT. Com valores numéricos, os espaços, carriage RETURN (CR) e fim de linha (line feed) são ignorados. O primeiro caracter que não for um destes, será assumido como o começo do número. Este terminará no próximo espaço, fim de linha, carriage return ou vírgula.

Se o BASIC MSX está transferindo um arquivo de dados sequencial para uma string, espaços, carriage returns e fins de linha também são ignorados. O primeiro caracter encontrado que não seja um destes, é assumido para o começo de uma string. Se este primeiro caracter for aspas ("), a string consistirá por todos os caracteres lidos entre a primeira e a segunda aspas.

Então, uma string não pode conter aspas como caracter.

Se o primeiro caracter da string não for aspas, esta estará sem cotação (começo/fim), e terminará na primeira vírgula, carriage return ou fim de linha que for lido (ou após 295 caracteres serem gravados). Se um "fim de linha" for encontrado, quando um item numérico ou string está sendo carregado, o item será terminado.

```
EXEMPLO: 10 OPEN "EXEMPL2.DAT" FOR INPUT AS # 1
          20 INPUT # 1, A$
          30 PRINT A$
          40 IF EOF (1) = 0 THEN 20
          50 CLOSE # 1
          60 END
```

### KILL

SINTAXE: KILL "< arqespec > "

PROPÓSITO: Deleta um arquivo do disco

COMENTÁRIOS: Se um KILL for utilizado com um arquivo que ainda está aberto (OPEN), um erro "ARQUIVO ABERTO" ocorrerá.

KILL é usado para todos tipos de arquivos de disco:  
Programas, arquivos aleatórios e arquivos de dados sequenciais.

Exemplo: 200 KILL "DADO 1. DAT"

#### LINE INPUT #

SINTAXE: LINE INPUT # < num arq > , < variável string >

PROPÓSITO: Lê uma linha completa (até 254 caracteres), sem limitações, de um arquivo de disco sequencial para uma variável string.

COMENTÁRIOS: < num arq > é o número pelo qual o arquivo foi aberto. < variável STRING > é o nome da variável em que a linha será transferida.

LINE INPUT # lê todos os caracteres do arquivo sequencial até um carriage RETURN. O próximo LINE INPUT # lê todos os caracteres até o próximo carriage return. (Se uma sequência line feed/carriage return for encontrada, esta sequência é interpretada como um fim de string com um caracter line feed (fim de linha)).

LINE INPUT # é usado especialmente em casos que a linha de dados de um arquivo foi quebrado em partes, ou se um programa MSX BASIC foi gravado em ASCII e está sendo carregado por outro programa. (Veja "SAVE.")

```
Exemplo: 10 OPEN "LIST" FOR OUTPUT AS # 1
          20 LINE INPUT "Informação de Cliente?"; C$
          30 PRINT #1, C$
          40 CLOSE 1
          50 OPEN "LIST" FOR INPUT AS # 1
          60 LINE INPUT # 1, C$
          70 PRINT C$
          80 CLOSE 1
```

RUN

Informação de cliente? José Silvério 3355

José Silvério 3355

OK

#### LOAD

SINTAXE: LOAD < nome arq > (.R)

PROPÓSITO: Carrega um arquivo do disco na memória.

COMENTÁRIOS: < nome arq > é o nome do arquivo quando este foi gravado.

A opção R roda o programa automaticamente, após ele ser carregado.

LOAD fecha todos os arquivos que tenham sido abertos e deleta todas as variáveis e linhas de programa corrente na memória, antes do comando carregar o programa designado.

Se a opção R for usada com LOAD, o programa será rodado, após ser carregado e todos os arquivos de dados abertos serão mantidos abertos. Então, LOAD com a opção R pode ser usado para ligar vários programas (ou segmentos do mesmo programa). informações podem ser trocadas entre os programas, usando seus arquivos de dados em disco.

Até o programa designado ser achado e ser carregado o programa residente na máquina é mantido.

```
EXEMPLO: LOAD "STRK", R
          LOAD "B: MEUARO"
```

#### LSET e RSET

SINTAXE: LSET < variável string > = < expressão string >  
RSET < variável string > = < expressão string >

PROPÓSITO: Desloca dados da memória para um buffer de arquivo aleatório. (Em preparação para um PUT)

COMENTÁRIOS: Se < expressão string > requer menos caracteres que os designados pelo FIELD para a < variável string >, LSET "left - justifies" o string no campo e RSET "right - justifies" a string. (Espaços são usados para bloquear as posições extras). Se a string for muito grande para o campo, caracteres são deslocados da direita.

Valores numéricos devem ser convertidos para strings, antes de serem usados com LSET ou RSET. (Veja "MKI\$, MKS\$, MKD\$, ")

```
EXEMPLO: 150 LSET A$ = MKS$(AMT)
          160 LSET D$ = DESC($)
```

Obs: LSET e RSET podem também ser usados com uma variável string não mapeada para "left - justify" ou "right - justify" uma string em um dado campo. Por exemplo:

```
110 A$ = SPACE$(20)
120 RSET A$ = N$
```

Este comando "right - justify" a string N\$ em um campo de 20 caracteres. Isto pode ser muito prático para formatação de dados para impressora.

## MAX FILES

**SINTAXE:** MAXFILES = < expressão >

**PROPÓSITO:** Especifica o número máximo de arquivos abertos em um determinado momento.

**COMENTÁRIOS:** < expressão > pode variar de 0 a 15. Quando "MAXFILES = 0" for executado, só SAVE e LOAD poderão ser elaborados.

O número máximo de arquivos de disco é 6, ainda que a < expressão > deste estatuto seja mais que 6.

## MERGE

**SINTAXE:** MERGE < NOMEARQ >

**PROPÓSITO:** Une o arquivo de disco especificado ao programa corrente na memória.

**COMENTÁRIOS:** < nomearq > é nome do arquivo, quando este foi gravado. O arquivo deve estar gravado em código ASCII. (se não, um erro "Acesso incorreto" ocorrerá).

Caso os números das linhas do arquivo de disco não coincidam com os números das linhas do programa, estas substituirão automaticamente as linhas correspondentes na memória.

O DDX BASIC sempre volta para o nível do comando, após a execução de um MERGE.

**EXEMPLO:** MERGE "NUMBER"

## NAME

**SINTAXE:** NAME < arqantig > AS < novoarq >

**PROPÓSITO:** Troca o nome de um arquivo de disco.

**COMENTÁRIOS:** < arqantig > deve existir e < novoarq > não deve existir; de outra forma, um erro ocorreria. Após o comando NAME, o arquivo reside no mesmo disco, na mesma área, mas com novo nome.

Se nenhum drive for especificado, o drive corrente será acionado.

**EXEMPLO:** NAME "PERA" AS "FRUTA"

Neste exemplo, o arquivo que havia sido nomeado PERA agora consta como FRUTA.

Caso os números das linhas do arquivo de disco coincidam com os números das linhas do programa, aquelas substituirão automaticamente as linhas correspondentes na memória.

## OPEN

**SINTAXE:** OPEN" < nomearq > (FOR < mode > ) AS (##) < numarq > (LEN= < reclen > )

**PROPÓSITO:** Permite E/S para um arquivo de disco.

**COMENTÁRIOS:** Um arquivo de disco deve ser aberto, antes que qualquer operação de E/S possa ser efetuada.

OPEN aloca um buffer para E/S ao arquivo e determina o modo de acesso que será usado com o buffer.

< mode > pode ser as seguintes opções:

FOR OUTPUT especifica o modo de saída sequencial.

FOR INPUT especifica o modo de entrada sequencial.

FOR APPEND especifica o modo de adição sequencial, após o fim de um arquivo existente.

PADRÃO especifica o modo E/S aleatória.

< numarq > consiste em um número inteiro cujo valor está entre 1 e o valor máximo de números de arquivos determinados por MAXFILES. O número é, então, associado ao arquivo, enquanto este permanecer aberto e é usado como referência de outros comandos de E/S a serem aplicados no arquivo.

< nomearq > é o nome do arquivo a ser relacionado como opcional para E/S do programa corrente.

< reclen > é uma expressão integral que, se incluída, determina a quantidade de bytes para arquivos aleatórios. A quantidade padrão de bytes é 256, este é o maior número possível; o menor é 1.

O número máximo de arquivos de disco é 6, mesmo se MAXFILES determina um número maior que 6.



OBS. Se o modo de entrada ou edição sequenciais for usado com um arquivo inexistente, o erro "Arq. não existe" ocorrerá. Se o modo de saída sequencial for usado com um arquivo já existente, o arquivo antigo será deletado.

Um arquivo pode ser aberto para entrada sequencial ou acesso aleatório em mais de um número de arquivo ao mesmo tempo. Um arquivo pode ser aberto para saída, porém em só um número de arquivo ao mesmo tempo.

EXEMPLO: 10 OPEN "INVEN" FOR INPUT AS # 1

PRINT # e PRINT # USING

SINTAXE: PRINT # < numarq >, ( USING < expstring >; ) < Lista de expressões >

PROPÓSITO: Grava dados em um arquivo de disco.

COMENTÁRIOS: < numarq > é o número associado ao arquivo, quando este foi aberto para saída. (expstring) consiste em caracteres formatados conforme descritos em "PRINT USING".

As expressões na < lista de expressões > são expressões numéricas e/ou STRING que serão gravadas no arquivo.

PRINT # não comprime dados do disco. Uma imagem dos dados é grava no disco, exatamente como seria mostrada na tela, com um PRINT. Por essa razão, deve-se tomar cuidado para delimitar os dados para o arquivo de disco. Na < lista de expressões >, expressões numéricas devem ser separadas por ponto e vírgula (;). Ex: ▀

PRINT # 1,A;B;C;X;Y;Z

(Se forem usadas vírgulas como separação, os espaços em branco que existirem entre os campos de impressão também serão gravados no disco).

Expressões string devem ser separadas por ponto e vírgula. Para formatar as expressões string exatamente como desejado no disco, use separações explícitas na < lista de expressões >.

Por exemplo , LET A\$="CAMERA" AND B\$="3846-4"

O estatuto:

PRINT # 1,A\$;B\$

Gravará CAMERA 3846-4 no disco. Como não há separações, este exemplo não pôde ser entrado como duas strings separadas. Para corrigir o problema, entre com separações explícitas como a seguir:

PRINT # 1,A\$;" ";B\$

A imagem gravada será:

CAMERA, 3846-4

que pode ser carregar novamente em duas variáveis string separadas. Se as strings contêm vírgulas, ponto e vírgulas, espaços consideráveis, fim de linha ou carriage return, escreva estas delimitadas por aspas.

Por exemplo: LET A\$="CAMERA, AUTOMÁTICA" and B\$="3846-4"

PRINT # 1,A\$;B\$

Gravará a seguinte imagem no disco:

CAMERA, AUTOMÁTICA 3846-4

E o estatuto:

INPUT # 1,A\$,B\$

Entrará com "CAMERA" para a\$ e "AUTOMÁTICA 3846-4" para B\$. Para separar estes strings corretamente no disco, escreva dupla aspas na imagem de disco, usando CHR\$(34) como a seguir:

PRINT # 1,CHR\$(34);CHR\$(34);CHR\$(34);CHR\$(34)

Este estatuto grava a seguinte imagem no disco:

"CAMERA, AUTOMÁTICA" "3846-4"

E o estatuto

INPUT # 1, A\$,B\$

Carregará "CAMERA, AUTOMÁTICA" em A\$ e "3846-4" em B\$

O PRINT # pode também ser usado com a opção USING, para controlar o formato do arquivo do disco. Por exemplo:

PRINT # 1, using "\$\$ ## # #. ## # ";J;K;L

## PUT

**SINTAXE:** PUT (#) < numarq > (, < num registro > )

**PROPÓSITO:** Grava um registro de um buffer aleatório em um arquivo de disco aleatório.

**COMENTÁRIOS:** < numarq > é o número pelo qual o arquivo de disco foi aberto. Se < num registro > for omitido, será assumido o próximo número de registro. (Após o último PUT), o maior número de registro é 4.294.967.295; o menor é 1.

**EXEMPLO:**

```
10 OPEN "EXEMPL. DAT" AS # 1
20 FIELD # 1, 2 AS A$, 10 AS B$
30 FOR I%=1 TO 10
30 FOR J%=1 TO 10
40 INPUT N$, S$
50 LSET A$ = MKI$(N%)
60 LSET B$ = S$
70 PUT # 1, I%
80 NEXT
90 CLOSE # 1
100 END
```

**OBS.:** LSET ou RESET devem ser usados para colocar caracteres no buffer de arquivo aleatório, antes de executar um PUT.

Qualquer tentativa de ler ou gravar, após o fim do buffer, causa um erro "campo maior".

## RUN

**SINTAXE:** RUN < nomearq > (.R)

**PROPÓSITO:** Carrega um arquivo de um disco na memória e roda-o.

**COMENTÁRIOS:** < nomearq > é o nome usado, quando o arquivo foi gravado.

RUN fecha todos os arquivos abertos e deleta todo o conteúdo da memória, antes de carregar o programa designado. Porém, com a opção R, todos os arquivos de dados permanecem abertos.

Caso < nomearq > não seja especificado, este comando roda o programa residente na memória.

**EXEMPLO:** RUN "NOVOARQ",R

## SAVE

**SINTAXE:** SAVE < arquespec > (,A)

**PROPÓSITO:** Grava um arquivo de programa no disco.

**COMENTÁRIOS:** < arquespec > é o nome do arquivo a ser gravado no disco. (Deve estar entre aspas). Caso este já exista no disco, o arquivo novo será sobreposto.

Use a opção A para gravar o arquivo em ASCII. De outra forma, o DDX-BASIC gravará o arquivo em formato comprimido em binário. ASCII toma maior espaço no disco, mas alguns acessos de disco requerem arquivos neste formato. Por exemplo, o comando MERGE e alguns comandos do sistema operacional, assim como LIST, podem requerer este formato ASCII.

Obs. "CSAVE" e "SAVE" são usados para gravar binário e ASCII em arquivos de casete. Mas "SAVE" e "SAVE ..., A" são usados em caso de arquivos de disco.

**EXEMPLO:** SAVE "COM2",A

## SYSTEM

**SINTAXE:** CALL SYSTEM ou \_SYSTEM

**PROPÓSITO:** Sai do Basic e volta para o DDX-DOS

**COMENTÁRIOS:** Este comando só é válido, quando o Basic foi inicializado pelo DDX-DOS.

Com este comando todos os arquivos são fechados e todos os programas e dados na memória são destruídos.

## VERIFY

**SINTAXE:** CALL VERIFY (ON/OFF)

**PROPÓSITO:** Liga/desliga o modo de verificação (leitura dos dados após gravação)

**COMENTÁRIOS:** O comando VERIFY ON liga o modo de verificação. Sempre que alguns dados forem

gravados no disco, estes serão lidos do disco e verificados. Caso esta verificação não esteja correta, um "ERRO DE E/S" ocorrerá.

O comando VERIFY OFF desliga o modo de verificação.

O modo padrão é VERIFY OFF.

OBS.: A gravação demora mais tempo, mas é mais garantida usando o modo de verificação.

### 2.3.2 FUNÇÕES

CVI, CVS, CVD  
DSKF  
DSKIS  
EOF  
INPUT  
LOC  
LOF  
MKIS, MKSS, MKDS  
VARPTR

#### CVI, CVS, CVD

SINTAXE: CVI (string de 2bytes)  
CVS (string de 4bytes)  
CVD (string de 8bytes)

PROPÓSITO: Converte STRINGS em valores numéricos. Valores numéricos que foram carregados de um arquivo de disco aleatório devem ser convertidos novamente em números. CVI converte uma string de 2bytes em um inteiro. CVS converte uma string de 4bytes em um número de precisão simples. CVD converte uma string de 8bytes em um número de dupla precisão.

EXEMPLO: 70 FIELD # 1,4 AS N\$, 12 AS B\$, ...  
80 GET # 1  
90 y = CVS (N\$)

#### DSKF

SINTAXE: DSKF (<numdrive>)

PROPÓSITO: Verifica a área livre no disco especificado em KBYTES.  
O número do drive corresponde ao nome do drive da seguinte forma:

0 drive titular  
1 drive A:  
2 drive B:

EXEMPLO: PRINT DSKF (1)

#### DSKIS

SINTAXE: DSKIS ( < numdrive > , < num-setor-lógico > )

PROPÓSITO: Lê o setor especificado da memória especificado pelo conteúdo de (OF351H, OF352H).

< num drive > é 0 para drive padrão, 1 para A, 2 para B.

< num-setor-lógico > é um número que em Q. Nenhuma verificação é feita no valor do número do setor.

OBS.: Esta área de memória é destruída, quando qualquer estatuto (FILES, OPEN, CLOSE, PRINT #, etc.) for executado.

#### EOF

SINTAXE: EOF ( < NUMARQ > )

PROPÓSITO: Saber se o fim de um arquivo sequencial foi encontrado. Retorna-1 (verdade), se assim for. Use EOF para testar o fim de arquivo, enquanto estiver carregando dados, para evitar erros de "FIM DE ARQUIVO".

O arquivo especificado pelo < numarq > deve ser aberto como entrada sequencial.

**EXEMPLO:**

```

10 OPEN "DATA" FOR INPUT AS # 1
20 C=0
30 IF EOF (1) THEN 100
40 INPUT # 1,M(C)
50 C=C+1 : GO TO 30

```

**INPUT\$**

**SINTAXE:** INPUT\$(X,[# | Y])

**PROPÓSITO:** Lê dados do terminal ou do número do arquivo Y. Retorna uma string de X caracteres. Todos os caracteres de controle serão saltados exceto CONTROL-STOP, que é usado para interromper a execução de uma função INPUT\$.

**EXEMPLO:**

```

5   Liste o conteúdo de um arquivo sequencial em hexadecimal.
10  OPEN "DATA" FOR INPUT AS # 1
20  EOF (1) THEN 50
30  PRINT HEX$(ASC(INPUT$(1, # 1)));
40  GO TO 20
50  PRINT
60  END

```

**LOC**

**SINTAXE:** LOC( < numarq > )  
onde < numarq > é o número pelo qual o arquivo foi aberto.

**PROPÓSITO:** Com arquivos de disco aleatório, LOC dá o número de registros lidos ou gravados por um GET ou PUT.

Se o arquivo foi aberto, mas nenhuma E/S de disco foi executado ainda, LOC recebe 0. Com arquivos sequenciais, LOC dá o número de registros lidos ou gravados no arquivo desde que foi aberto. Quando nenhum registro foi lido arquivo sequencial, desde que foi aberto, LOC recebe 1, porque o SYSTEM leu o primeiro setor.

**EXEMPLO:** 200 IF LOC(1)>5 SO THEN STOP.

**LOF**

**SINTAXE:** LOF( < numarq > )

**PROPÓSITO:** LOF fornece o tamanho do arquivo especificado em bytes.

**EXEMPLO:** IF NUM% > LOF(1) THEN PRINT "Inválido"

**MKIS, MKS\$, MKD\$**

**SINTAXE:** MKIS( < exp. inteira > )  
MKS\$( < exp. de precisão simples > )  
MKD\$( < exp. de dupla precisão > )

**PROPÓSITO:** Converte valores numéricos em strings. Qualquer valor numérico, que é colocado em um buffer de arquivo aleatório com um LSET ou RESET, deve ser convertido em uma string. MKIS Converte um inteiro em uma string de 2 bytes. MKS\$ converte um número de precisão simples em uma string de 4bytes. MKD\$ converte um número de dupla precisão em um string de 8bytes.

**EX.:**

```

90  AMT = (K+T)
100 FIELD #1,8 AS D$,20 AS N$
110 LSET D$ = MKS$(AMT)
120 LSET D$=MKD$(AMT)
130 PUT # 1

```

•  
•  
•  
Veja também "CVI,CVS,CVD,"

## VARPTR

**SINTAXE:** VARPTR ( # < numarq > )

**PROPÓSITO:** VARPTR fornece o endereço do bloco de controles do arquivo designado numarq.

**EXEMPLO:** 100 x =USR (VARPTR ( # 1 )

### 2.3.3 Códigos e Mensagens de erro

NÚMERO do código	MENSAGEM	DESCRIÇÃO DO ERRO
50	CAMPO MAIOR	Um FIELD está tentando designar mais caracteres do que os possíveis para um registro de um arquivo aleatório.
51	ERRO INTERNO	Ocorreu um mal funcionamento interno no BASIC. Notifique o fabricante sob quais condições a mensagem apareceu.
52	N. DO ARQUIVO	Um estatuto ou comando refere-se a um arquivo cujo número não foi aberto ou está fora do limite dos números de arquivo especificados na inicialização.
53	ARQ. NÃO EXISTE	Um LOAD, KILL ou OPEN refere-se a um arquivo que não existe no drive corrente.
54	ARQUIVO ABERTO	Um modo de saída sequencial OPEN é determinado para um arquivo que ainda está aberto; ou um KILL é determinado para um arquivo que está aberto.
55	FIM DO ARQUIVO	Um INPUT é executado após o arquivo já ter sido entrado por inteiro ou para um arquivo nulo (vazio). Para evitar este erro, use EOF para detectar o fim do arquivo.
56	NOME ARQUIVO	Um formato ilegal é usado para nomear um arquivo com um LOAD, SAVE, KILL ou OPEN (nome do arquivo com mais de 8 caracteres).
57	COMANDO DIRETO/ARQUIVO	Um estatuto direto é encontrado, quando carregando um arquivo em ASCII. A carga é terminada.
58	ARQUIVO SEQUENCIAL	Um GET ou PUT é usado em um arquivo sequencial.
59	FALTA 'OPEN'	Um estatuto de entrada ou saída é executado em um arquivo não aberto.
60	ERRO NA FAT	O disco não está inicializado.
61	ACESSO INCORRETO	É feita uma tentativa de usar um PUT, GET ou LOF com um arquivo sequencial, LOAD com um arquivo aleatório, ou executar um OPEN com um modo diferente de "FOR INPUT", "FOR OUTPUT", "FOR APPEND" ou padrão (aleatório).
62	DRIVE INVÁLIDO	Um nome de arquivo inválido é especificado.
63	SETOR INVÁLIDO	O arquivo não foi fechado.
64	ARQUIVO JÁ ABERTO	O nome do arquivo especificado em um estatuto NAME é idêntico a um já em uso no disco.
65	ARQUIVO EXISTENTE	Todo o espaço do disco está ocupado.
66	DISCO CHEIO	É feita uma tentativa de criar um novo arquivo, usando SAVE ou OPEN, quando todos os 255 espaços p/diretório estão ocupados.
67	DIRETÓRIO CHEIO	
68	DISCO PROTEGIDO	Um comando de gravação é executado em um disco protegido contra gravação.
69	ERRO DE E/S	Um erro de entrada ou saída de dados ocorre em uma operação de E/S com disco.
70	DISCO DESCONECTADO	Não há disco no drive especificado.
71	ERRO AO RENOMEAR	Tentativa de renomear um arquivo com um nome já existente no disco.

