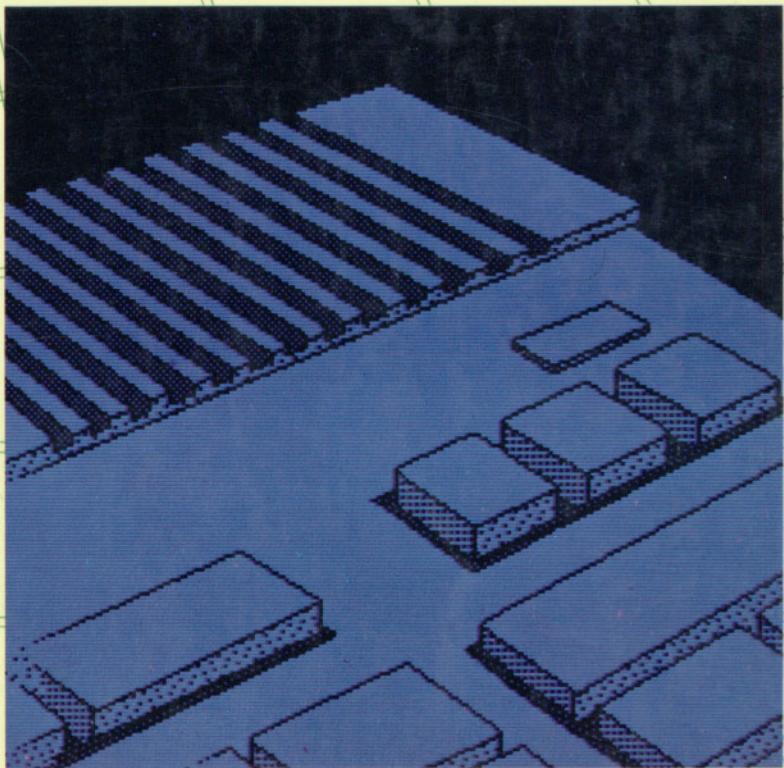


**COLEÇÃO MSX**



**COMO USAR  
SEU HOTBIT**

**MSX**

6  
COLEÇÃO MSX

# COMO USAR SEU HOTBIT

LUIZ TARCÍSIO DE CARVALHO JR.  
PIERLUIGI PIAZZI  
ROBERTO WATANABE  
RENATO DA SILVA OLIVEIRA



**SHARP**

## EXPEDIENTE

Coordenação Editorial

Pierluigi Piazzì

Coordenação Didática

Betty Fromer Piazzì

Editoração

L. Tarcísio de Carvalho Jr.

Arte

Ana Lúcia Antico

Produção

Rosa K. Fromer



### ALEPH

Publicações e Ass. Pedagógica Ltda.

C.P. 20.707 CEP 01498

São Paulo SP

Tel: (011) 813-2033

### EPCOM

Equipamentos Eletrônicos Ltda.

R. José Carlos de Macedo Soares s/n

Bloco B - Pavilhão 4

Taboão da Serra SP

Dados de Catalogação na Publicação (CIP) Internacional  
(Câmara Brasileira do Livro, SP, Brasil)

Carvalho Júnior, Luiz Tarcísio de, 1955—

C329c      Como usar seu HOTBIT / Luiz Tarcísio de Carvalho  
Jr. ... ]et al.[. — São Paulo : Aleph, 1987.  
(Coleção MSX)

1. HOTBIT (Computadores) 2. MSX (Computadores)  
I. Título.

87-0985

CDD-001.64

Índices para catálogo sistemático:

1. HOTBIT : Computadores : Processamento de dados 001.64
2. MSX : Computadores : Processamento de dados 001.64

# SUMÁRIO

<b>Capítulo 1</b>	
COMO INSTALAR SEU HOTBIT .....	007
<b>Capítulo 2</b>	
HOTBIT E SEUS APLICATIVOS .....	019
<b>Capítulo 3</b>	
INTRODUÇÃO AO HOT-LOGO .....	027
<b>Capítulo 4</b>	
INTRODUÇÃO AO HOT BASIC .....	079
<b>Capítulo 5</b>	
HOT DATA — Gerenciador de Dados .....	132
<b>Capítulo 6</b>	
HOT PLAN — Planilha de Cálculos .....	149
<b>Capítulo 7</b>	
HOT WORD — Processador de Textos .....	155
<b>Apêndice I</b>	
Como Usar o Gravador Cassete .....	166
<b>Apêndice II</b>	
Caracteres do HOTBIT .....	172
<b>Apêndice III</b>	
Mensagens de Erro .....	176
<b>Apêndice IV</b>	
Funções Matemáticas .....	180
<b>Apêndice V</b>	
Uso da Impressora .....	182

# COMO INSTALAR SEU HOTBIT

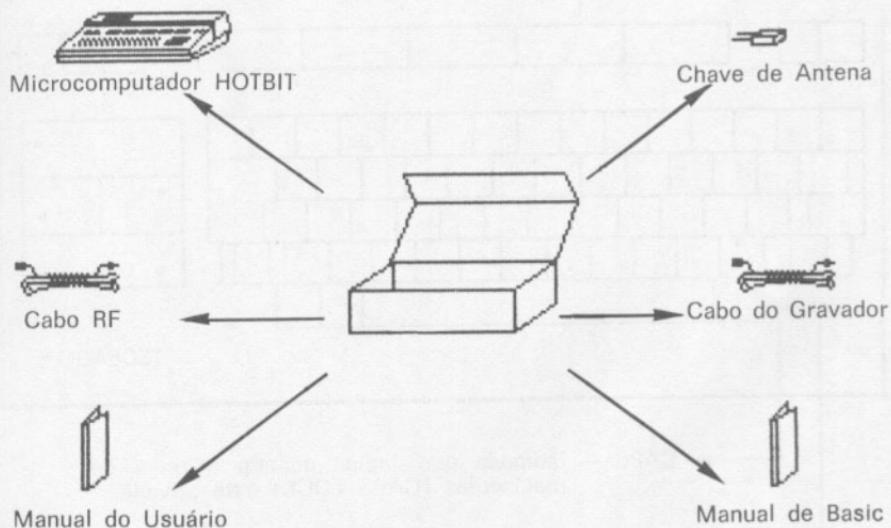
Para que seu HOTBIT esteja instalado e pronto para uso é indispensável a leitura cuidadosa deste capítulo. Com isso você evitará danos ao aparelho e poderá utilizá-lo adequadamente.

Inicialmente um conselho: NÃO LIGUE O HOTBIT NA TORMADA antes de completar a leitura destas instruções passo a passo.

## PRIMEIRO PASSO: DESEMBALAGEM

Retire o aparelho da caixa (figura 1.1) manuseando-o com cuidado para que não caia e nem receba pancadas. Verifique se todos os itens relacionados na figura estão presentes. Você está recebendo um verdadeiro sistema de computação e não uma simples máquina. Se faltar algum item reclame imediatamente com o revendedor.

Figura 1.1 — Relação dos componentes.



## SEGUNDO PASSO: RECONHECIMENTO

Segurando o HOTBIT com cuidado olhe para as figuras seguintes para se familiarizar com os principais conectores e interruptores.

Lembre-se: é um hábito bem brasileiro não se ler os folhetos e manuais de instruções. Nesse caso, porém, não se trata de um simples radinho ou vídeo-game: você está lidando com um microcomputador, equipamento que exige um processo de aprendizagem por parte do usuário para que possa ser utilizado, a contento.

Figura 1.2 — Parte Superior

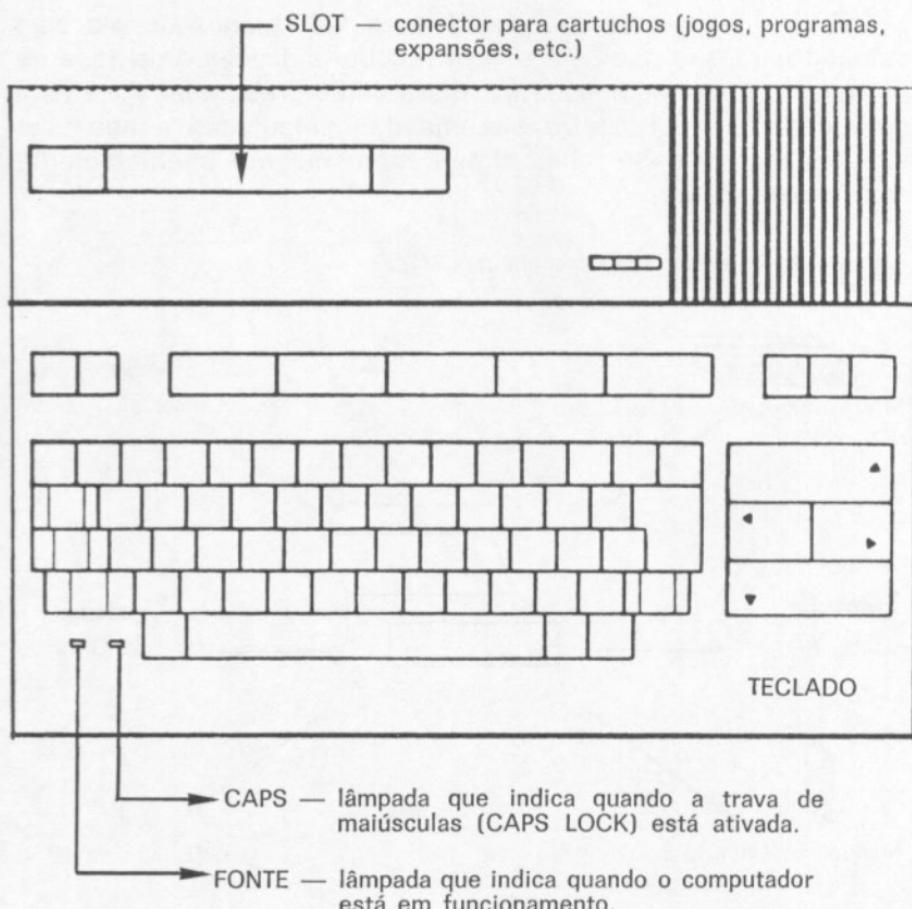


Figura 1.3 — Parte Traseira

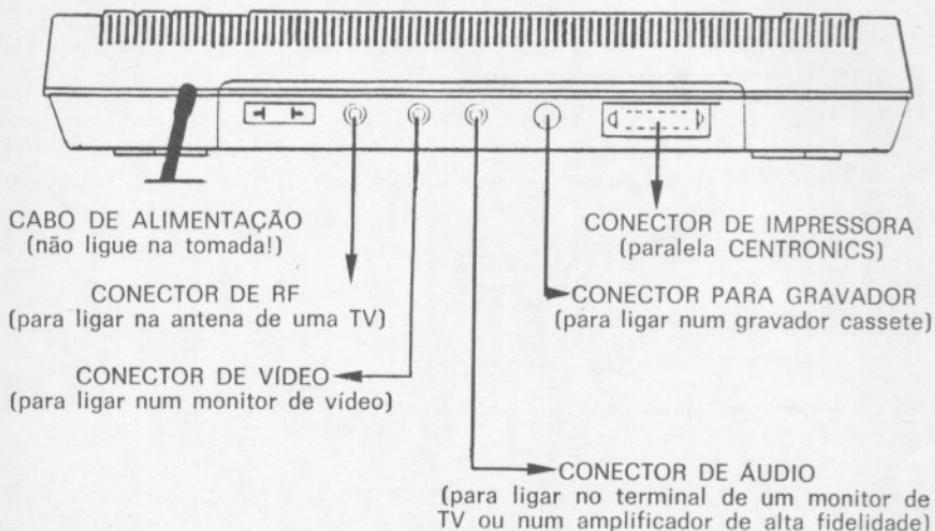


Figura 1.4 — Vistas Laterais

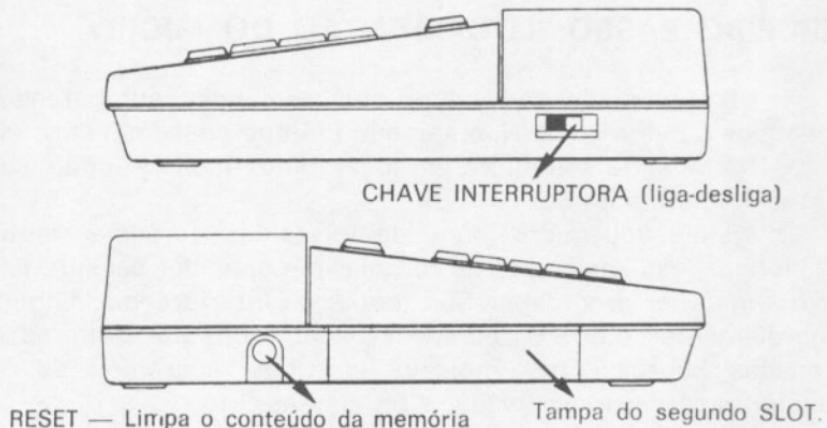


Figura 1.5 — Vista Frontal

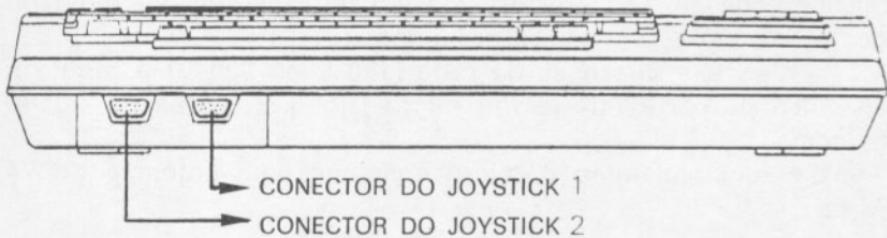
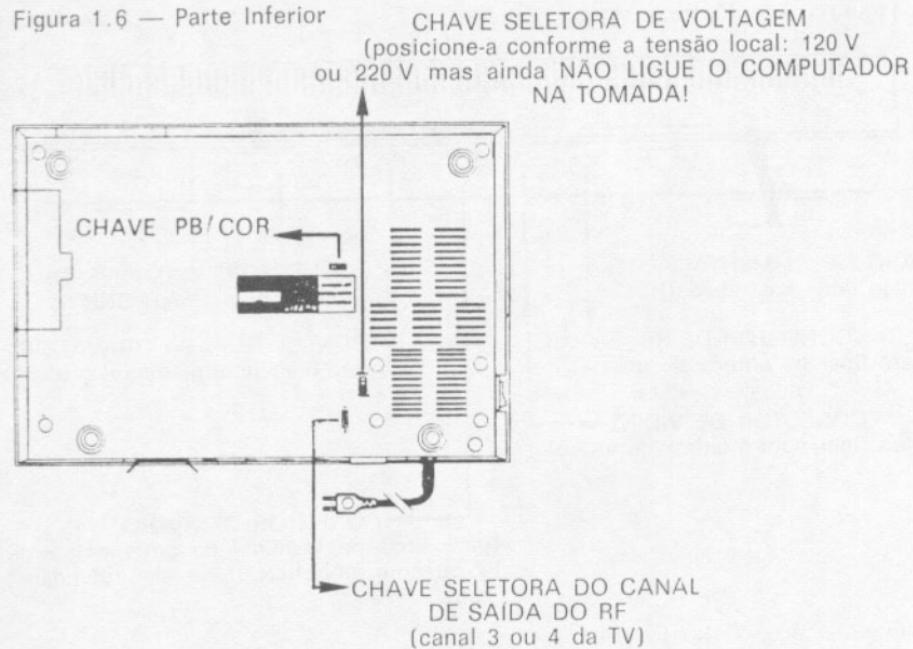


Figura 1.6 — Parte Inferior



### TERCEIRO PASSO: LOCALIZAÇÃO DO MICRO

- \* Não manipule seu micro com violência, submetendo-o a choques ou vibrações. Não coloque objetos pesados sobre ele.
- \* Não instale seu micro em local muito quente, úmido, com poeira ou onde bata sol.
- \* Instale seu micro longe de fortes interferências eletromagnéticas, especialmente se for usar um gravador cassete para ler e armazenar programas. São fontes de interferência: bobinas e enrolamentos elétricos como os encontrados em reatores de lâmpadas fluorescentes, motores elétricos, aparelhos de TV, transformadores, ímãs fortes, alto-falantes, etc.

O próprio micro tem, em seu interior, um mini-transmissor de TV que pode ser regulado no canal 3 ou 4. Esse transmissor pode causar interferência em um aparelho de rádio ou de TV, se estiverem muito próximos.

\* Verifique a tensão da rede (120 V ou 220 V) e regule a chavinha na parte inferior do micro (fig. 1.6) antes de ligá-lo na tomada.

Em funcionamento o micro consome uma potência de 24 watts.

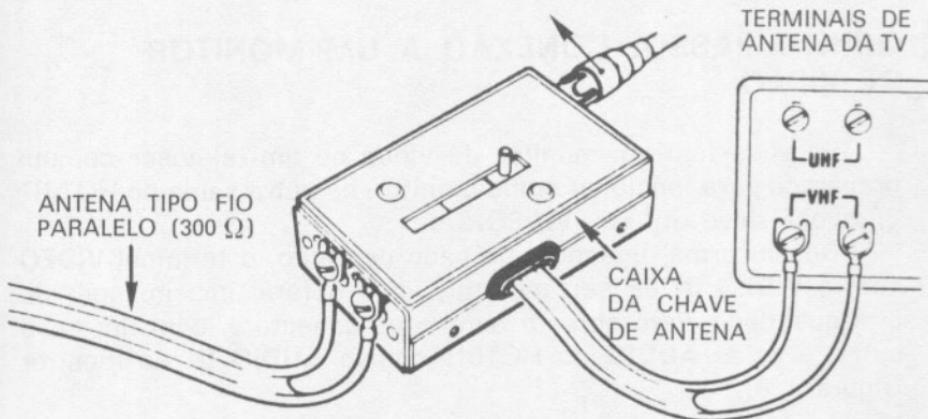
## QUARTO PASSO: CONEXÃO A UM TELEVISOR COMUM

Seu HOTBIT pode ser ligado a um televisor comum, a cores ou preto e branco.

Procure escolher um canal que não seja usado em sua cidade e sintonize o televisor nesse canal (3 ou 4), usando o seletor de canais do televisor. Coloque a chave seletora situada na parte inferior do micro (fig. 1.6) no canal correspondente e ligue os cabos conforme a figura 1.7.

Figura 1.7 — Ligação do HOTBIT aos terminais de antena da TV com fio paralelo.

PARA O COMPUTADOR

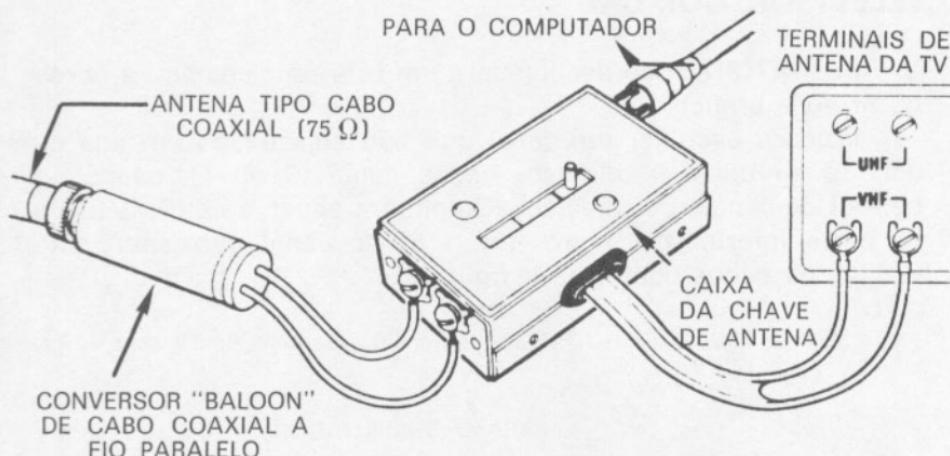


Dessa forma, comutando-se a chave de antena, o televisor pode receber tanto a programação comum quanto o sinal emitido pelo pequeno transmissor do micro (no canal 3 ou 4 conforme a escolha feita anteriormente).

Se a sua antena não tem um fio paralelo tipo "talharini" ( $300 \Omega$ ) mas um cabo coaxial "rolizo", você precisa de um "balloon" conversor para ligar a antena externa à chave seletora (figura 1.8).

Se você conectou seu HOTBIT a um televisor como foi descrito, pode pular o quinto e passar a ler o sexto passo.

Figura 1.8 — Ligação do HOTBIT aos terminais de antena da TV com cabo coaxial.

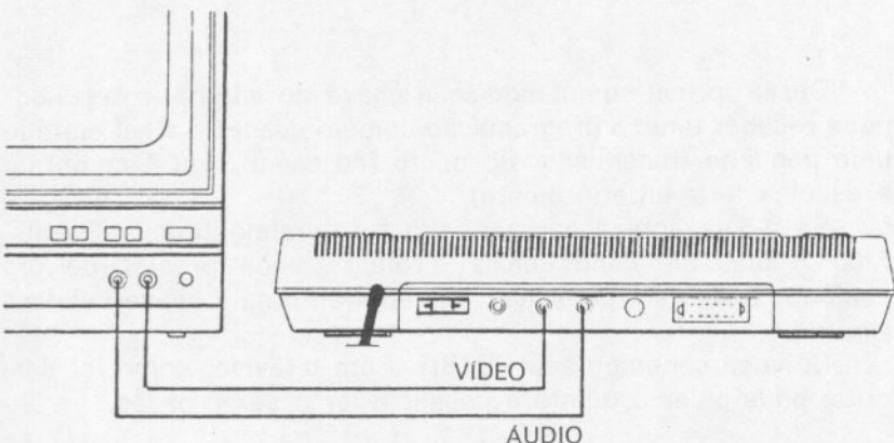


## QUINTO PASSO: CONEXÃO A UM MONITOR DE VÍDEO

Se você tem um monitor de vídeo ou um televisor comum preparado para funcionar como monitor, há outra saída do HOTBIT que você deve utilizar: **VÍDEO**.

Dessa forma, ligando pelo cabo de vídeo, o terminal **VÍDEO** com o **VÍDEO IN** de seu monitor, você obterá uma imagem de alta qualidade. Para obter o som você conectará, com um cabo extra, a saída **ÁUDIO** do HOTBIT com o **ÁUDIO IN** do monitor (figura 1.9).

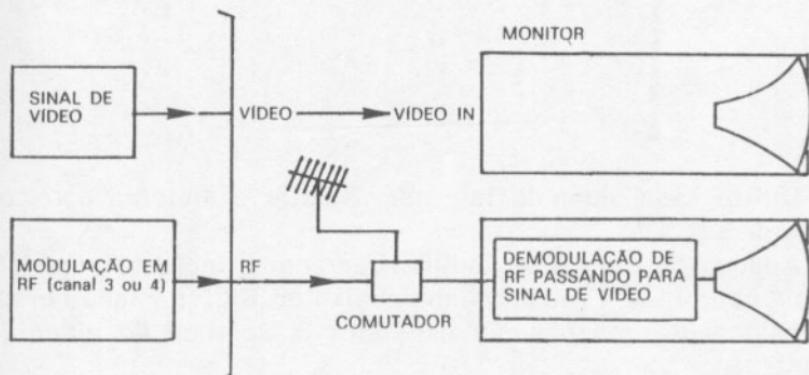
Figura 1.9 — Ligação do HOTBIT a um monitor de vídeo.



Se o seu monitor não tem entrada de áudio, você pode conectar a saída **AUDIO** do HOTBIT com o amplificador de seu equipamento de som.

Ligar o HOTBIT num monitor de vídeo implica numa melhor qualidade de imagem. Analisando a figura 1.10, note que o sinal gerado pelo computador sofre menos "processamento", ficando livre de uma série de possíveis interferências.

Figura 1.10 — A diferença entre um sinal de vídeo e de RF.



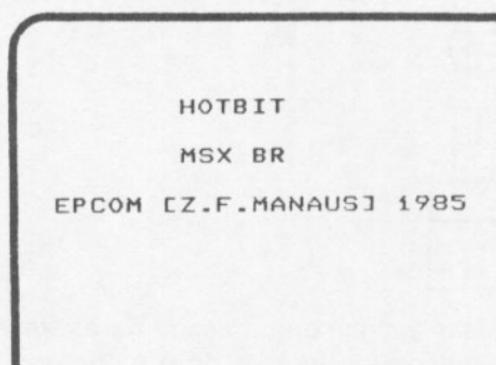
Quando você se familiarizar mais com seu HOTBIT poderá inclusive fazer uma série de experiências com um gravador de vídeo-cassete.

## SEXTO PASSO: USANDO O TECLADO

Agora você pode ligar seu HOTBIT na tomada.

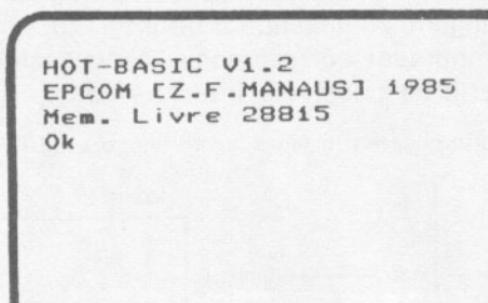
Ao ligar seu HOTBIT à tela de TV (ou seu monitor) surge a mensagem da figura 1.11.

Figura 1.11 — A primeira tela do HOTBIT.



A seguir, a tela muda para o que é mostrado na figura 1.12.

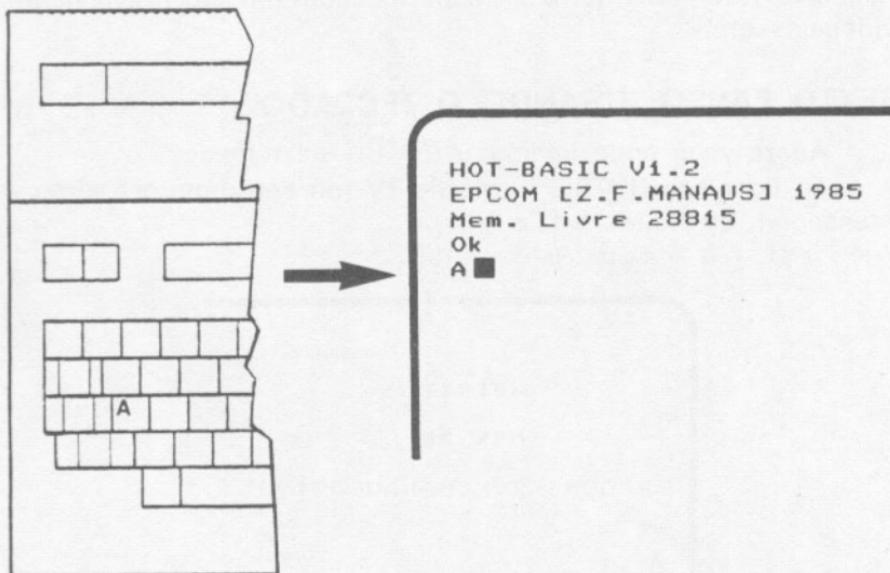
Figura 1.12 — A segunda tela do HOTBIT.



Utilize essa segunda tela para ajustar a sintonia e o contraste de sua TV.

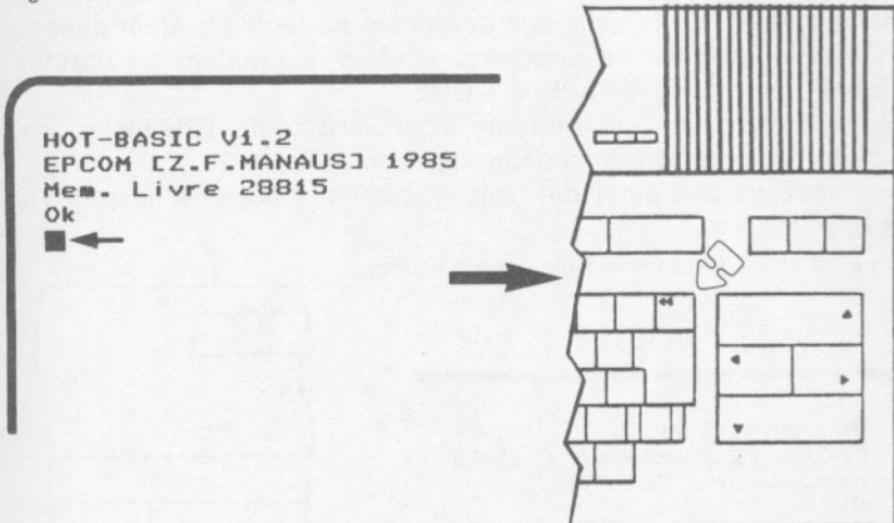
Apenas para você se familiarizar com o teclado, pressione a tecla da letra A. O quadradinho abaixo do **Ok** (chamado cursor) se desloca de uma posição e a letra A aparece no vídeo (figura 1.13).

Figura 1.13 — Pressione a tecla A.



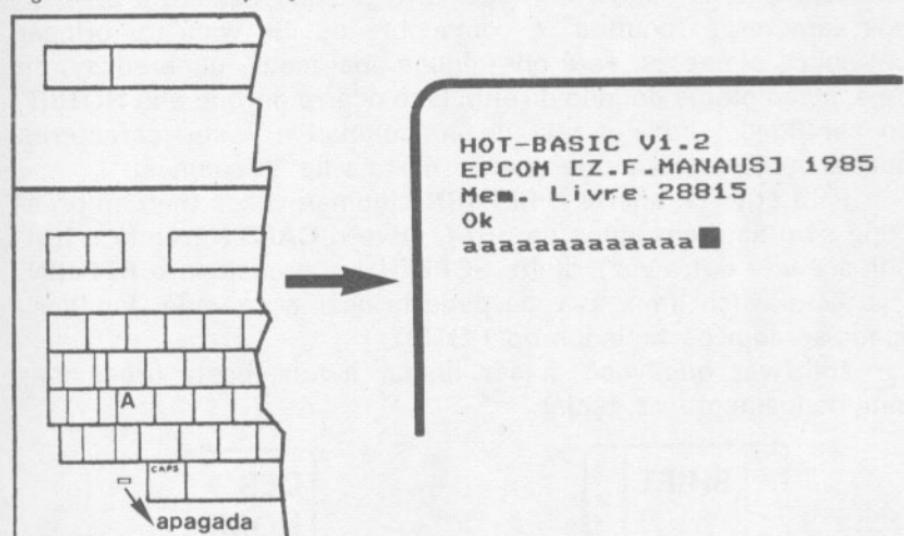
Se você quiser apagar o que acaba de escrever, basta pressionar a tecla indicada na figura 1.14 e o cursor volta apagando.

Figura 1.14 — O uso da tecla "borracha".



Pressione agora a tecla CAPS, destravando as minúsculas — a luzinha **CAPS** apaga-se — e digite novamente a tecla A, mantendo-a pressionada algum tempo. Como você pode observar na figura 1.15, a letra agora é minúscula (pois o **CAPS** foi desativado) e foi escrita uma porção de vezes. Realmente, mantendo-se uma tecla pressionada por algum tempo, ativa-se um processo de repetição automática.

Figura 1.15 — A repetição automática e a trava de maiúsculas (CAPS).

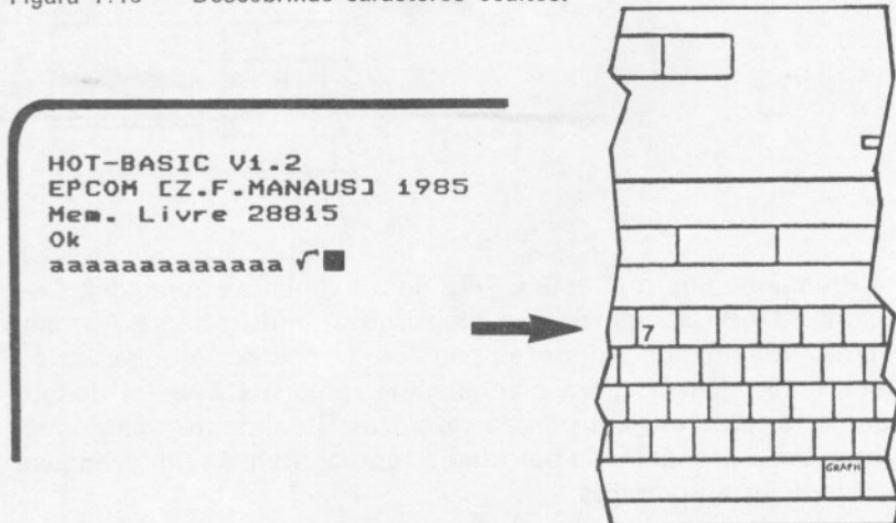


Seu HOTBIT permite gerar no vídeo todos os caracteres maiúsculos e minúsculos que aparecem no teclado. Além desses existem muitos outros caracteres ocultos que podem ser obtidos através das teclas **GRAPH** e **CODE**.

Por exemplo, experimente apertar a tecla **GRAPH** e, enquanto a mantém pressionada, digite o número 7.

Você verá o sinal de "raiz quadrada" aparecer na tela (figura 1.16).

Figura 1.16 — Descobrindo caracteres ocultos.

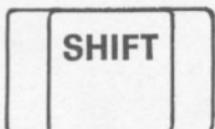


Nas figuras seguintes você vê o teclado básico, a posição dos caracteres "ocultos" e como obtê-los. Se você for brincar um pouco com eles, verá que alguns aparecem "quebrados", ou seja, incompletos do lado direito. Isso ocorre porque seu HOTBIT, ao ser ligado, ativa a tela de 40 colunas e esses caracteres aparecem completamente apenas na tela de 32 colunas.

Para ativá-la, aperte o **RETURN** algumas vezes (não se preocupe com as mensagens de erro), ative o **CAPS** (fazendo a luzinha acender outra vez), digite SCREEN 1 e pressione o **RETURN**.

A tela fica limpa e você pode brincar a vontade, familiarizando-se com os teclados do HOTBIT.

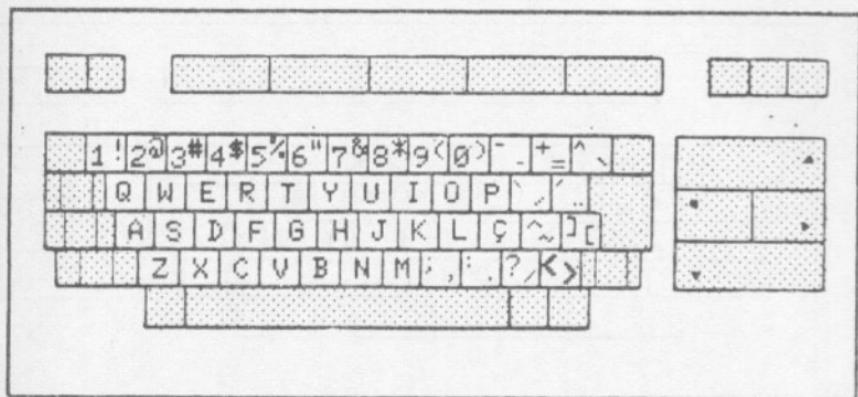
Toda vez que você quiser limpar a tela, basta pressionar simultaneamente as teclas



+

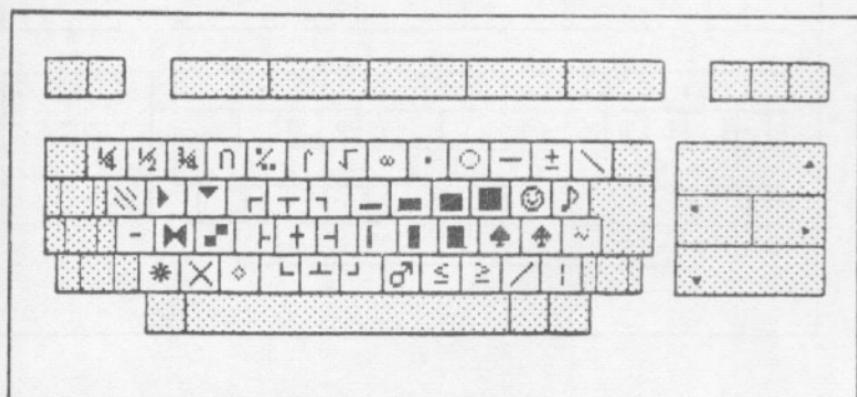


Figura 1.17 — Os caracteres básicos.



Normal : Letras minúsculas e símbolos inferiores.  
Com SHIFT : Letras maiúsculas e símbolos superiores.  
CAPS : Trava em maiúsculo só as letras.  
Observação : As teclas de acentuação só são executadas após a pressão da vogal.

Figura 1.18 — Caracteres obtidos pressionando a tecla GRAPH.



\* As teclas sombreadas são descritas no capítulo 4.

Figura 1.19 — Caracteres obtidos pressionando "SHIFT + GRAPH"

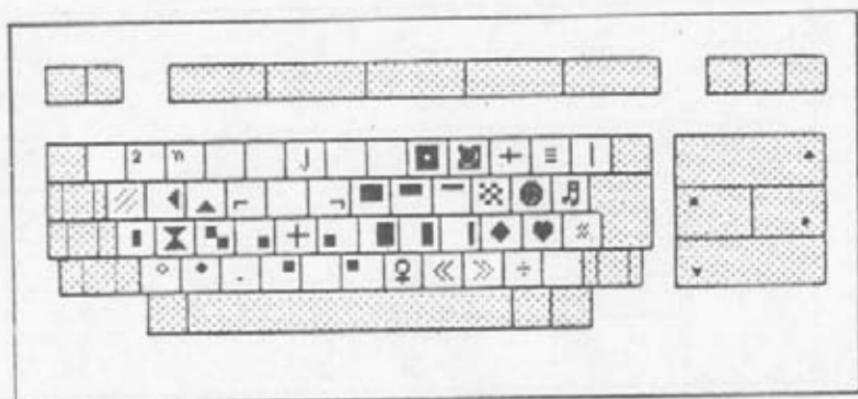
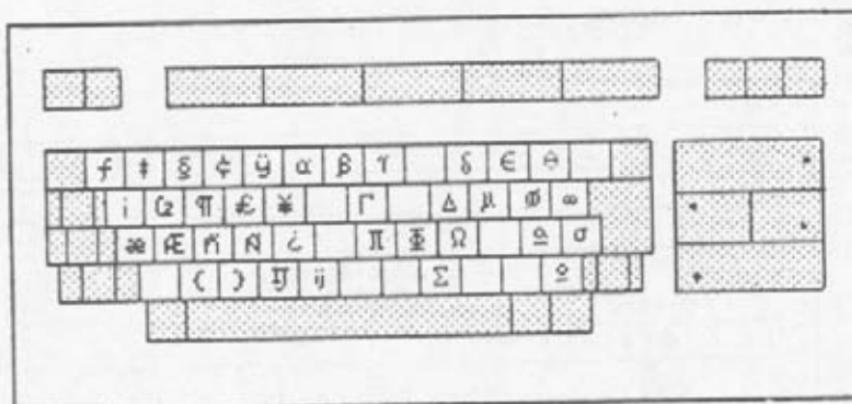
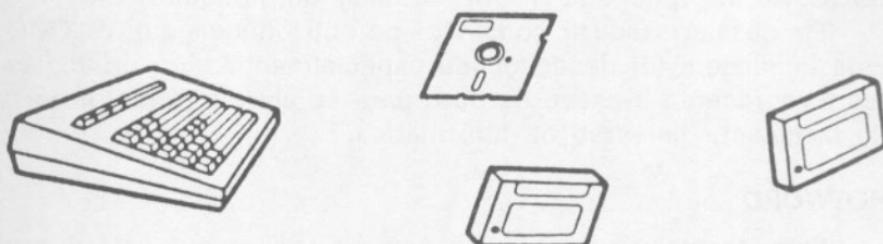


Figura 1.20 — Caracteres obtidos pressionando "SHIFT + CODE"



\* As teclas sombreadas são descritas no capítulo 4.

# HOTBIT E SEUS APLICATIVOS



## HOTBIT HB-8000

Este é um microcomputador construído segundo o padrão internacional MSX.

Além do teclado e das conexões, que já foram descritos no capítulo 1 deste manual, ele tem em seu interior alguns dispositivos que devem ser conhecidos.

O microprocessador é seu cérebro principal: no padrão MSX utiliza-se o Z-80, o mais moderno, veloz e eficiente microprocessador de 8 bits já fabricado.

O VDP (Vídeo Display Processor) é um cérebro auxiliar que gerencia as funções de vídeo do HOTBIT, permitindo os incríveis efeitos gráficos a 16 cores do seu micro. Ele é assistido por uma memória RAM (ou seja, memória na qual podemos escrever e ler) de 16 Kbytes (cada Kbyte equivale a 1024 caracteres).

O PSG é o outro cérebro auxiliar que gerencia os efeitos sonoros do HOTBIT. Ele permite produzir ruídos, sons e música em 3 vozes!

Além disso, você tem 64 Kbytes de memória RAM para armazenar as informações e os programas.

Seu micro já vem de fábrica com 32 Kbytes de memória ROM gravados.

A ROM é uma memória que podemos ler mas na qual não podemos escrever. Nela estão gravadas as instruções do Sistema Operacional e o Interpretador Basic.

Este Interpretador Basic é um verdadeiro tradutor que transforma as instruções dadas pelo usuário em linguagem Basic (de fácil aprendizado, como você verá no capítulo 4) para a complicada Linguagem de Máquina que só o Z-80 entende.

## **HOTLOGO**

Neste cartucho você tem um interpretador de linguagem LOGO. Ao utilizá-lo, seu HOTBIT torna-se um poliglota!

Ele passa a traduzir comandos de outra linguagem: o LOGO. Esta linguagem foi desenvolvida especialmente para principiantes e representa a chave de ouro para se abrir a primeira porta do fascinante universo da Informática.

## **HOTWORD**

Este cartucho transforma seu HOTBIT num versátil processador de texto. Ligando-se o micro a uma impressora de boa qualidade, damos um passo gigantesco para o futuro, fazendo a máquina de escrever ficar tão obsoleta quanto uma vitrola de 78 r.p.m.!

O texto é escrito, apagado, reescrito e corrigido na tela do computador.

Somente após ficar do jeito que você quiser, é que ele é impresso ou gravado em fita (ou disquete).

É o fim do corretor e da borracha! E se você quiser produzir um texto parecido com um pré-gravado, por exemplo, um contrato no qual você quer mudar um nome e uma cláusula, basta carregar o texto anterior e fazer apenas as correções necessárias, sem precisar digitar tudo de novo!

## **HOTDATA**

Neste cartucho está contido um gigantesco fichário no qual você pode fazer desde um controle de estoque até uma agenda.

Em relação a um fichário de papel ele tem a vantagem de ter uma certa inteligência própria: você pode pedir, por exemplo, a relação de todos os seus clientes que fazem aniversário no mês que vem e obtê-la em questão de segundos!

## **HOTPLAN**

Se alguma vez você já fez uma complicada seqüência de cálculos e foi obrigado a refazer tudo porque apareceu, no meio do caminho uma mudança na taxa de câmbio ou um Plano Cruzado, você é um sério candidato para o cargo de usuário fanático do HOTPLAN.

Com este cartucho você transforma seu HOTBIT numa enorme planilha de cálculos que permite desde a confecção de um

orçamento doméstico até a elaboração da folha de pagamento de uma pequena empresa.

## **COMO SE INICIAR EM COMPUTAÇÃO**

Os capítulos seguintes deste livro podem ser lidos em qualquer ordem, em função de seu interesse e de seus conhecimentos prévios sobre a computação.

Se você tem interesse em se iniciar na linguagem BASIC, aconselha-se a leitura atenta do capítulo 4.

Após essa leitura introdutória, se você quiser uma referência mais completa sobre o BASIC-MSX, aconselha-se a leitura do "Manual de Basic" editado pela EPCOM.

Querendo aprofundar seus conhecimentos sobre o BASIC e a arquitetura da máquina, aconselha-se a seguinte literatura:

- \* MSX — Guia do Usuário — Paul Hoffman — Editora McGraw-Hill — Discute exaustivamente o BASIC-MSX e orienta o leitor no uso do DISK BASIC e MSX DOS (para uso de discos magnéticos).
- \* APROFUNDANDO-SE NO MSX — Piazzi, Oliveira, Maldonado et al. — Editora Aleph — Discute com detalhes as características da arquitetura da máquina. Indicado para quem domina o BASIC e quer começar a elaborar programas em Linguagem de Máquina.
- \* COLEÇÃO DE PROGRAMAS PARA MSX vol 1 e 2 — Oliveira et al. — Editora Aleph — Jogos, desenhos, músicas e utilitários em BASIC-MSX. Indicado para principiantes na linguagem BASIC.
- \* MSX — PRÁTICA E DOMÍNIO — Nélson Casari — Editora Atlas — Programas comentados em BASIC-MSX. Indicado para principiantes.

- \* JOGOS MSX vol. I e II — Young e Burd — Editora McGraw-Hill — Coletânea de programas listados em BASIC-MSX.
- \* MSX — GUIA DO OPERADOR — Young e Burd — Editora McGraw-Hill — Manual de referência rápida dos comandos e funções do BASIC-MSX.
- \* JOGOS DE HABILIDADE — Martins, Sung, Dias e Guazelli — Editora Aleph — Coletânea de programas em BASIC e Linguagem de Máquina para MSX.
- \* PROGRAMAÇÃO AVANÇADA EM MSX — Figueiredo, Maldonado e Rossetto — Editora Aleph — Para os programadores mais experientes que querem usar recursos descobertos no livro "APROFUNDANDO-SE NO MSX", utilizando a velocidade da Linguagem de Máquina.

Se você quer aprender a programar numa linguagem mais acessível que o BASIC, aconselha-se começar pela leitura do capítulo 3.

Nesse capítulo, você terá uma introdução ao LOGO, linguagem desenvolvida especialmente para crianças e de incrível potencial em aplicações didáticas. Algumas pesquisas mostram que quem se "alfabetizou" em LOGO, tem muito mais facilidade para aprender outras linguagens, incluindo o BASIC.

Nesse caso, aconselha-se, depois de ler o capítulo 3, a leitura do livro:

- \* HOTLOGO — Godoy, Lacerda, Lepíscopo e Mendes — Editora Aleph — Livro didático especialmente escrito para crianças, mas indicado também para programadores principiantes nessa linguagem.

Numa primeira etapa do seu contato com a computação, você pode decidir não começar pela programação. Há a possibilidade de se iniciar na utilização de um aplicativo útil para usar seu HOTBIT como uma máquina de escritório bastante sofisticada.

Se você dispõe de uma impressora e quer utilizar seu HOTBIT como uma sofisticada máquina de escrever, conecte o cartucho HOT WORD e comece pela leitura do capítulo 7.

Da mesma forma, se você quiser começar criando um super-fichário eletrônico, conecte o cartucho do HOT DATA e leia o capítulo 5.

Se você já ouviu falar em planilhas eletrônicas e quiser se iniciar no mundo da informática aprendendo a lidar com uma delas, conecte o cartucho HOT PLAN e leia atentamente o capítulo 6.

Antes porém, um lembrete: CONECTE OU DESCONECTE OS CARTUCHOS SEMPRE COM SEU HOTBIT DESLIGADO!

Como o uso desses três cartuchos é descrito de maneira resumida nos capítulos 5 (HOT DATA), 6 (HOT PLAN) e 7 (HOT WORD), aconselha-se a leitura dos livros:

\* HOTDATA — R. M. Watanabe — Editora Aleph.

\* HOTPLAN — R. M. Watanabe — Editora Aleph.

\* HOTWORD — R. M. Watanabe — Editora Aleph.

Nos livros acima citados, todas as potencialidades desses três programas são descritas de maneira detalhada e extremamente didática.

Apesar de ser possível o uso de fita cassete para arquivar os dados produzidos por esses programas, aconselha-se o uso de um disk drive de maneira a gerar arquivos em disco magnético. Apenas dessa forma o uso dos programas HOT DATA, HOT PLAN e HOT WORD pode ser considerado realmente profissional.

Para se familiarizar com o uso de um disk drive, aconselha-se a leitura do livro:

\*: USANDO O DISK DRIVE NO MSX — Rubens Pereira Jr.  
— Editora Aleph — Descrição e uso das funções e comandos do DSK BASIC e dos sistemas operacionais HB-DOS e HB-MCP.

Qualquer que seja a sua opção, lembre-se da extrema versatilidade do seu computador e não se limite a uma ou duas aplicações ou linguagens.

O MSX é um padrão extremamente elástico, permitindo a conexão com uma gama vastíssima de periféricos, de maneira a implementar as mais variadas aplicações.

Relaciona-se a seguir alguns periféricos já existentes no mercado e sua utilidade quando conectados a seu HOTBIT.

**\* GRAVADOR CASSETE** — Qualquer gravador cassete de boa qualidade pode ser conectado ao HOTBIT para gravar programas e arquivos seqüenciais. O HB-2400 é um gravador cassete fabricado pela SHARP especialmente para o uso com o HOTBIT, permitindo alta confiabilidade na gravação de dados em fita.

**\* IMPRESSORA** — Qualquer impressora com conexão paralela padrão CENTRONICS pode ser conectada ao HOTBIT. Certifique-se de que a impressora tem a tabela de caracteres MSX ou ABICOMP para poder imprimir os caracteres acentuados.

Existem no mercado interfaces que permitem a conexão do HOTBIT com máquinas de escrever eletrônicas.

**\* DISK DRIVE** — Existem no mercado acionadores e controladores de disco padrão MSX. O acionador de disco deve ser padrão IBM-PC e a interface controladora padrão MSX, ou seja, ela deve trazer uma complementação da linguagem BASIC residente no micro. Isso torna seu HOTBIT apto a gerenciar discos.

A SHARP lançou seu HB-3600 que permite operar o disco com o DSK BASIC e, além disso, permite o uso de dois sistemas operacionais: o HB-DOS, compatível com o MSX-DOS, e o HB-MCP, compatível com o CP/M.

A unidade acionadora de discos que deve operar com a interface citada é a HB-6000, também fabricada pela SHARP.

É o periférico indispensável para usos profissionais de seu computador que permite rodar uma quantidade enorme de software desenvolvida anteriormente para outros computadores.

**\* JOYSTICK** — Apesar de estar associado à imagem de "video-game", o joystick é um periférico muito útil não só para jogos como para utilizar determinados softwares para desenho.

**\* MOUSE** — Além de funcionar como "paddle", o mouse conectado ao seu HOTBIT pode ser usado como substituto do Joystick.

**\* CARTUCHO DE EXPANSÃO** — É um cartucho que contém memória RAM e, ao ser conectado ao micro, amplia sua memória disponível. Quando o HOTBIT está ligado a um controlador HB-3600 acoplado a um acionador HB-6000 e está sob a gerência do HB-MCP, reconhece o cartucho como drive F. Isso significa que, apesar de ter um único drive, o usuário pode operar como se

tivesse dois, sendo um deles de acesso extremamente veloz. Muito útil, por exemplo, para quem utiliza o dBase II.

\* **RAM DISK** — É um banco de memória que substitui ou complementa o disk drive e é de acesso extremamente rápido.

\* **EXPANSOR DE SLOTS** — É uma espécie de "benjamim" que permite a conexão de vários cartuchos e/ou expansões. O padrão MSX permite o endereçamento de 16 X 64 kbytes, ou seja, 1 Megabyte. Através do expanson de slots toda essa potencialidade pode ser utilizada.

\* **PLACA DE 80 COLUMNAS** — Permite ligar o HOTBIT a um monitor de vídeo, produzindo uma tela de 80 colunas. Indispensável para o uso de certos programas que exigem 80 caracteres por linha na tela.

\* **MODEM** — Dispositivo que conecta o seu HOTBIT a uma linha telefônica, permitindo comunicação micro a micro, acesso a VÍDEO-TEXTO e CIRANDÃO, acesso a computadores de bancos, uso do micro como pseudo-telex, etc.

\* **PLOTTER** — Trata-se de um dispositivo de desenho no qual uma caneta se movimenta para traçar desenhos, comandada pelo micro. No Brasil já existe um modelo de plotter de baixo custo conectável à saída CENTRONICS (da impressora) que permite produzir desenhos em folhas de papel padrão A4 (21 cm x 30 cm).

Essa plotter pode usar uma caneta de "ponta porosa" ou caneta "nankim" para desenhos técnicos e tem um alfabeto residente com dez tamanhos de letras, selecionáveis por software.

Esse periférico é muito importante para usar seu HOTBIT em desenhos técnicos (minicAD).

Como você pode perceber, seu HOTBIT é apenas o começo de um sistema que pode se tornar extremamente sofisticado e completo.

Para encerrar este capítulo indicamos a relação das editoras cujos livros foram citados:

**ALEPH** — Av. Brig. Faria Lima, 1451 — conj. 31  
01451 — São Paulo — SP

**ATLAS** — R. Cons. Nébias, 1384  
01203 — São Paulo — SP

**McGRAW-HILL** — R. Tabapuã, 1105  
04533 — São Paulo — SP

Para maiores informações sobre os periféricos citados e sobre outros softwares disponíveis no mercado contacte:

**Núcleo de Apoio ao Usuário — SHARP**  
**Estrada do Campo Limpo, 6197**  
**CEP: 05787 — São Paulo — SP**  
**Telefone: (011) 842-9461**

# INTRODUÇÃO AO HOT-LOGO

Informática, computador, programação. Essas palavras causam, muitas vezes, uma sensação de intimidação, distância, e não raro, um clima de "guerra nas estrelas".

Para a maioria dos educadores e pais que se preocupam em preparar seus filhos para o futuro, os computadores são encarados como "algo" alienante e temeroso. A rápida e recente expansão da indústria de computadores colocou à nossa disposição e uso, todo o potencial dessa "máquina estranha". É compreensível, portanto, que não estando familiarizados com os recursos computadorizados, existam reservas quanto o futuro um tanto impessoal que os computadores produzirão no cotidiano de nossas crianças.

Os computadores, por terem ampla abrangência, realizam um número de tarefas muito maior que as demais máquinas. Conseqüentemente, é válido supervalorizar seus desempenhos e imaginar estruturas complexas e de difíceis compreensão.

Entretanto, como os computadores estão mais facilmente disponíveis nos dias de hoje, é mais simples nos familiarizarmos com eles e atuarmos para detectar modos por meio dos quais melhoraremos nossa qualidade de vida.

Saymour Papert, professor do Instituto de Tecnologia de Massachusetts, criou uma linguagem de programação denominada LOGO, que torna o computador acessível até para crianças de baixa faixa de idade. A proposta de Papert é levar a criança a ensinar, com toda sua vivência, o computador.

Papert imaginou que cada criança terá num futuro, um computador à sua disposição. Ele criou o LOGO na esperança de que a criança manipulará seu computador como se fosse um poderoso lápis.

A SHARP, por sua vez, criou o HOT-LOGO, que é uma linguagem compatível com o LOGO idealizado por Papert.

A grande facilidade de acesso ao computador por meio do HOT-LOGO vem, em parte, da sua própria simplicidade.

A linguagem consiste de um vocabulário composto por palavras básicas em português tais como parafrente, paratrás, paradireita e paraesquerda.

Para executar os comandos do HOT-LOGO dispomos da TARTARUGA. Ela é uma entidade gráfica e luminosa, que vive na tela do computador e que pertence ao Universo HOT-LOGO.

Essa TARTARUGA fará os desenhos que ensinarmos, pois ela pode se movimentar pela tela e deixar um rastro por onde ela passar.

## COMO INSTALAR O HOT-LOGO

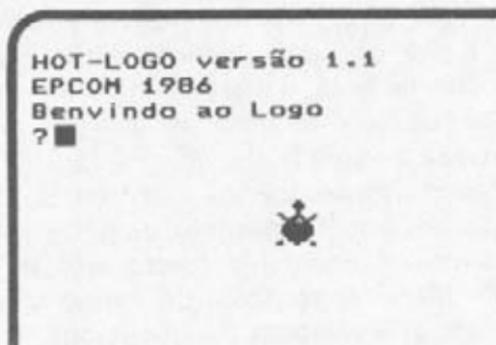
Para usar o HOT-LOGO, siga as instruções a seguir:

1 — Com o HOTBIT desligado, coloque o cartucho do HOT-LOGO num dos slots, mantendo sempre o furo do cartucho voltado para o lado esquerdo e a etiqueta voltada para você.

2 — Se você vai utilizar o gravador, verifique se ele está conectado ao computador.

3 — Ligue o computador e, depois de alguns instantes, a tela deverá se mostrar conforme a figura 3.1.

Figura 3.1 — Tela de apresentação do HOT-LOGO.



Essa tela indica que você está no universo HOT-LOGO. A tartaruga que aparece no centro da tela está pronta para receber suas ordens.

O ponto de interrogação (?) é chamado sinal de prontidão e o pequeno quadrado que aparece piscando ao seu lado é o cursor que indica o lugar onde vai aparecer o caractere (letra, número, etc.), que você digitar.

**ATENÇÃO:** verifique se a luz verde referente à tecla CAPS está apagada. Se estiver acesa, pressione essa tecla. Isso deve ser feito pois o HOT-LOGO só nos permite utilizar letras minúsculas em seus comandos.

## CONVERSANDO COM A TARTARUGA

Vamos tentar conversar um pouco com aquela tartaruga que está no meio da tela e ver o que ela é capaz de fazer.

Para isso, vamos tentar alguns comandos. Agora lembre-se: pressione a tecla **RETURN** após a digitação de cada comando. Essa tecla faz com que o HOT-LOGO interprete a ordem que está sendo dada.

Experimente digitar:

**a1ô**

Pressione a tecla **RETURN** e obterá como resposta:

**Ainda não aprendi alô**

Essa mensagem significa que a palavra alô não pertence ao vocabulário HOT-LOGO, isto é, ainda não lhe foi ensinado como proceder com esse nome.

Agora comande:

**parafrente**

E obterá como resposta:

**Não há entradas suficientes ->  
para parafrente**

Isso significa que o comando primitivo **parafrente** necessita de uma entrada: o número de passos que a tartaruga deve se deslocar.

Apague a tela de texto comandando **att** e tente agora:

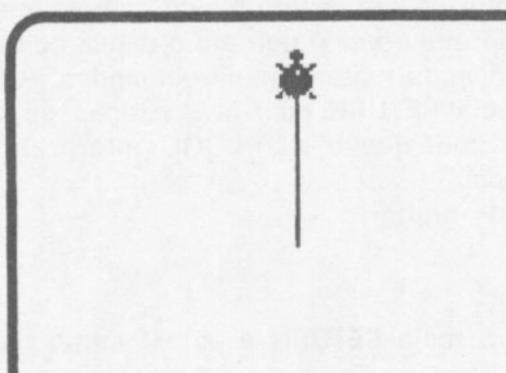
**parafrente 50  
ou  
pf 50**

Você deverá obter a tela da figura 3.2.

Da mesma maneira podemos deslocar a tartaruga para trás utilizando o comando primitivo **paratrás** (**pt**). Tente, por exemplo:

**paratrás 70  
ou  
pt 70**

Figura 3.2 — Tela após o comando para frente 50.



A tartaruga pode também girar. Ela “entende” os giros a serem dados como ângulos medidos em graus a partir da direção em que se encontra.

Se quisermos, por exemplo, girar a tartaruga para a direita 1/4 de volta (90 graus) devemos digitar:

```
paradireita 90  
ou  
pd 90
```

Observe que a tartaruga obedeceu ao comando **paradireita**, girando 1/4 de volta mas manteve a sua posição, isto é, não saiu do local onde estava.

Se você quiser girar a tartaruga para o lado contrário, utilize o comando **paraesquerda (pe)**. Por exemplo, para dar 1/2 volta (girar 180 graus) para a esquerda, digite:

```
paraesquerda 180  
ou  
pe 180
```

## DESENHANDO FIGURAS

Para iniciarmos uma nova figura vamos primeiro limpar a tela. Para isso, comande:

```
tat
```

Esse comando é a abreviação de "tartaruga apague tudo". Quando executado, a tela gráfica e de texto são limpas e a tartaruga é colocada em sua posição original.

Se quisermos apagar apenas o desenho e manter a posição e orientação da tartaruga usaremos:

```
ad  
ou  
apaguedesenho
```

Com alguns comandos, podemos desenhar até figuras complexas. Vamos no entanto, começar com algo mais simples como um quadrado.

Antes de começar, lembre-se de separar com um espaço comandos sucessivos e os comandos de seus parâmetros. Não se esqueça também de pressionar a tecla RETURN após a digitação dos mesmos.

Para obter um quadrado, comande a seqüência que se apresenta a seguir:

```
pf 90 pd 90  
pf 90 pd 90  
pf 90 pd 90  
pf 90 pd 90
```

Para obter o mesmo quadrado, você também pode simplesmente comandar:

```
repita 4 [pf 90 pd 90]
```

O comando primitivo **repita** permite ordenar ao HOT-LOGO um determinado número de instruções que se repetem. Esta lista de instruções deve vir sempre entre colchetes ([]).

Sempre limpando a tela após cada desenho com o comando **tat**, tente agora desenhar as figuras da tabela 3.1.

Note que para obtermos a circunferência, pedimos para que a tartaruga traçasse um polígono regular de 360 lados. É claro que não faz sentido falarmos em lados de circunferência. O efeito é apenas visual. Por ser um polígono regular de muitos e pequenos lados, ele se confunde com uma circunferência. Para fazer arcos e circunferências de outros tamanhos, varie o número de repetições, o valor do passo e do ângulo.

Experimente:

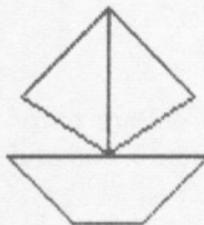
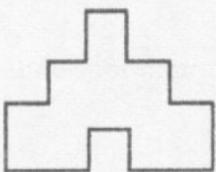
```
repita 180 [pf 0,5 pd 1]
e
repita 9 [pf 10 pe 10]
```

Tabela 3.1 — Figuras

FIGURA	COMANDO A SER DADO
TRIÂNGULO	repita 3[pf 50 pd 120]
POLÍGONO DE 6 LADOS	repita 6[pf 50 pd 60]
POLÍGONO DE 12 LADOS	repita 12[pf 20 pd 30]
CIRCUNFERÊNCIA	repita 360[pf 1 pd 1]

A limitação do HOT-LOGO é praticamente a sua imaginação. Procure fazer alguns desenhos como por exemplo os mostrados na figura 3.3.

Figura 3.3



Para que você tenha alguns recursos adicionais ao tentar fazer as suas figuras, é interessante conhecer mais alguns comandos:

1. Para andar com a tartaruga sem deixar rastro: **usenada (un)**.
2. Para voltar a deixar rastro: **uselápis (ul)**.
3. Para apagar linhas: **useborracha (ub)**. Para desativar **ub** use **ul**.
4. Para tirar a tartaruga da tela: **desapareçatat (dt)**.
5. Para que ela volte à tela: **apareçatat (at)**.

## PROGRAMANDO EM HOT-LOGO: PROCEDIMENTOS

O HOT-LOGO só reconhece uma série de comandos chamados primitivos, mas o seu vocabulário pode ser ampliado.

Programar em HOT-LOGO consiste exatamente em "ensinar à tartaruga" como proceder diante de uma nova palavra ou comando. Isto é, programar o HOT-LOGO é enriquecer o seu vocabulário.

Por exemplo, ao digitarmos

```
retângulo
```

o HOT-LOGO responderá:

```
Ainda não aprendi retângulo
```

Tecle agora,

```
aprenda retângulo  
ou  
ap retângulo
```

Dessa forma, entramos no modo de "Edição de Procedimentos" e enquanto não sairmos dessa modalidade o HOT-LOGO não executará as nossas ordens. Elas irão diretamente para a memória do computador onde serão armazenadas até desligarmos o HOTBIT ou pressionarmos o botão de RESET ou ainda se ordenarmos que o HOT-LOGO os esqueça.

A tela mostrará:

```
aprenda retângulo  
fim
```

Aperte a tecla RETURN, e obterá como resultado

```
aprenda retângulo  
■  
fim
```

Você pode, a partir deste ponto, digitar as suas instruções para desenhar um retângulo e a tela ficará assim:

```
aprenda retângulo  
repita 2 [pf 50 pd 90 pf 10 ->  
pd 90]  
fim
```

Tendo terminado as instruções, pressione a tecla ESC para sair da edição e o HOT-LOGO dará a seguinte mensagem:

```
retângulo aprendido  
?
```

Com esta mensagem o HOT-LOGO avisa que "aprendeu" o retângulo que imaginamos. O importante agora é que **retângulo** passou a fazer parte dos comandos do HOT-LOGO. Para testar, comande:

```
retângulo
```

O HOT-LOGO entendeu o que é **retângulo** e a tartaruga o desenhou na tela.

Nesse exemplo, **retângulo** é um comando secundário e recebe o nome de PROCEDIMENTO. Ele pode ser utilizado ou chamado sempre que você quiser.

Digite agora

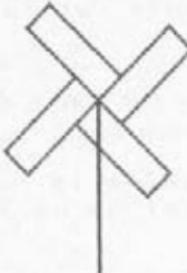
```
ap catavento  
pe 45  
repita 4 [retângulo pd 90°  
pd 45 pt 80  
fim
```

Comande então

```
tat  
catavento
```

Você deve obtido a figura 3.4. Isto mostra que o HOT-LOGO, além de ter aprendido **retângulo** e usá-lo agora como um SUB-PROCEDIMENTO, aprendeu também **catavento**.

Figura 3.4 — Catavento.

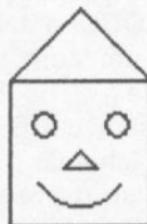


Como outras linguagens modernas, programas em HOT-LOGO consistem de pequenos e claros procedimentos cada um com um trabalho específico a executar. Vejamos um exemplo.

Deseja-se desenhar a cara do palhaço da figura 3.5. Depois de cuidadosamente planejada, podemos digitar:

figura 3.5

```
aprenda palhaço
cabeça
chapéu
olhos
nariz
boca
fim
```



Agora devemos ensinar ao computador como desenhar cada parte, como por exemplo:

```
aprenda cabeça
repita 4 [pf 50 pd 90]
fim
```

```
ap chapéu
repita 3 [pf 50 pd 120]
fim
```

Em seguida definem-se **olhos**, **nariz** e **boca**, porém antes de mandar executar o procedimento **palhaço** deve-se chamá-lo novamente com o comando

```
edite "palhaço"
ou
ed "palhaço"
```

Usando as setas e a tecla RETURN devemos inserir as linhas com as instruções de posicionamento. Assim, o procedimento **palhaço** ficará estruturado da seguinte maneira:

```
aprenda palhaço
cabeça
pf 50 pd 30 ←
chapéu
```

linha de posicionamento para que o chapéu fique acima da cabeça.

Assim, no HOT-LOGO, problemas podem ser divididos em pequenas partes. Procedimentos podem ser criados em separado para cada parte. Desse modo, podemos dividir trabalhos em procedimentos menores e depois combiná-los para formar SUPERPROCEDIMENTOS. Podemos ainda fazer interações entre os procedimentos.

Esse fato evidencia que HOT-LOGO é expansível, pois o usuário ensina ao computador novos comandos ou palavras.

O HOT-LOGO é também interativo, isto é, o programador pode digitar um comando ou programa e "ver" sua execução imediatamente.

Além disso, o HOT-LOGO é recursivo, isto é, um procedimento pode "chamar" a si mesmo ou fazer uma "cópia" dele próprio. Analise o procedimento a seguir:

```
aprenda cerca
haste
pd 90 pf 20 pe 90
cerca
fim
```

onde **haste** pode ser definido como

```
aprenda haste
retângulo
pf 50 pd 30
triângulo
pe 30 pt 50
fim
```

O procedimento **retângulo** já foi ensinado para fazermos o catavento, mas ainda falta definir **triângulo**. Desta forma,

```
aprenda triângulo
repita 3 [pf 10 pd 120]
fim
```

Agora limpe a tela com **tat**, comande **cerca** e veja o resultado.

Para interromper este procedimento basta pressionarmos simultaneamente as teclas CTRL e STOP.

## VARIÁVEIS

Até esse ponto, para fazermos a mesma figura de diferentes tamanhos, deveríamos "ensiná-las" separadamente. Assim, para obtermos um quadrado de lado 10 teríamos um procedimento,

para um quadrado de lado 20 "ensinariamós" outro procedimento e assim sucessivamente.

Imagine se tivéssemos a possibilidade de definirmos um procedimento que desenhasse figuras de tamanhos quaisquer.

O HOT-LOGO nos traz, para tornar possível essa idéia, um outro recurso: a variável.

A variável nada mais é do que uma "gaveta" reservada na memória do computador, onde depositamos informações.

Para ensinar um procedimento com entradas, isto é, utilizando variáveis, devemos obedecer certas regras. Para ilustrar, vejamos um exemplo.

```
aprenda caixa :lado
repita 4 [pf :lado pd 90]
fim
```

Observe que logo depois do procedimento **caixa** colocamos **:lado** que é o nome da "gaveta" onde será armazenado o número correspondente ao tamanho do lado de **caixa**. É exatamente o sinal **:** (dois pontos) que indica ao HOT-LOGO que esta palavra define uma variável e não um procedimento.

Pressionando a tecla ESC voltaremos ao modo direto.

Para executarmos este procedimento, deveremos simplesmente digitar:

```
caixa 50
```

e obteremos como resultado um quadrado de 50 passos de tartaruga como lado.

Para que tenhamos quadrados de diferentes tamanho, basta mudarmos a entrada, isto é, o número que será armazenado na "gaveta" designada.

Digite então:

```
caixa 40
caixa 30
caixa 20
```

Usando a recursão e a variável, podemos criar um procedimento que desenha quadrados e que pare quando uma determinada condição for verdadeira. Por exemplo:

```
aprenda caixas :lado
se :lado >10 [pare]
caixa :lado
caixas :lado-5
fim
```

Experimente executar este procedimento e tente explicar o que aconteceu. Para isso, digite, após pressionar a tecla ESC, as restrições:

```
caixas 80
caixas 10
caixas 8
```

## MAIS TARTARUGAS

Trabalhamos até aqui, somente com uma tartaruga: a de número 0. O HOT-LOGO nos permite desenhos com até 30 tartarugas. Para isso precisamos chamá-las com o comando primitivo **atençãotartaruga (atat)** e dar-lhes ordens. Por exemplo, digite o seguinte programa:

```
atat 1 pd 45
atat 2 pd 90
atat 3 pd 135
atat 4 pd 180
atat 5 pd 225
atat 6 pd 270
atat 7 pd 315
atat todas at pf 50
```

### Observações:

1 — Os comandos **pd 225**, **pd 270** e **pd 315** poderiam ser substituídos respectivamente por **pd -135**, **pd -90** e **pd -45**.

2 — O comando **todas** chama todas as tartarugas, isto é, a lista **[0,...,29]**.

3 — No exemplo acima, aparecem as tartarugas apenas quando o comando **todas** é executado.

Se quisermos que as tartarugas 0, 2, 4 e 6 desenhem um círculo alternadamente e que as tartarugas 1, 3, 5 e 7 desenhem uma estrela nas mesmas condições, faremos assim:

```
ap coisa
rg tat
atat 1 pd 45
atat 2 pd 90
atat 3 pd 135
atat 4 pd 180
atat 5 pd 225
atat 6 pd 270
atat 7 pd 315
atat todas at un pf 50 ul
atat [e 2 4 6] cada [repita ->
5 [pf 30 pd 144]]]
atat [e 1 3 5 7] cada [pd 90 r->]
repita 360 [pf 0,25 pd 133
fim
```

Após "ensinar" **coisa** ao HOT-LOGO, digite:

```
coisa
```

e observe a figura apresentada.

O comando **cada** faz com que as tartarugas executem as instruções da lista seguinte, sequencialmente.

Perceba que no início do procedimento **coisa** foi usado o comando **rg**. Esse comando é muito poderoso, ele apaga a tela gráfica e leva as tartarugas para as suas condições iniciais. Para maiores detalhes consulte a listagem dos comandos no final deste capítulo.

## COLOCANDO A TARTARUGA EM MOVIMENTO

A tartaruga no seu estado original está com velocidade 0. Se quisermos que ela se movimente pela tela, basta mudar a sua velocidade, que pode variar de -128 até 128.

Experimente:

```
mudevel 25
```

Ela começará a se movimentar na direção que está indicada.

Apesar do movimento, não perdemos controle sobre ela, ou seja, os comandos podem ser dados normalmente. Experimente:

```
pd 30
pe 120
pf 200
```

Podemos ainda fazê-la apagar as linhas por onde ela passa usando o comando **useborracha (ub)** ou andar sem deixar rastro com **usenada (un)**.

## MEXENDO NO TEXTO

Você notou que no HOT-LOGO trabalhamos ao mesmo tempo com uma tela gráfica (onde desenhamos) e uma tela de texto (onde escrevemos). Para que possamos visualizar melhor o nosso desenho, podemos apagar a tela de textos com o comando primitivo **apaguetexto (att)**.

Podemos também fixar o número de linhas a serem usadas. Por exemplo:

```
mudetexto 20
```

Assim fixaremos as 4 últimas linhas para texto.

## AS CORES DO HOT-LOGO

O HOT-LOGO oferece 16 possibilidades de cores.

Para trocar a cor de fundo utilize o comando **mudecf** seguido do número da cor desejada (consulte a Tabela 4.1 da pág. 92 ). Por exemplo:

```
mudecf 1
```

Para trocar a cor do lápis use **mudecl** seguido do número da cor. Exemplo:

```
mudecl 2 pf 20 mudecl 4 pf 2->  
0 mudecl 6 pf 20
```

Para alterar a cor da tartaruga comande **mudect** e o número da cor. Exemplos:

```
mudect 2  
mudect 10  
mudect 13
```

Podemos também pintar figuras. Assim, se desenharmos um quadrado na tela e posicionarmos a tartaruga no seu interior, podemos pintá-lo com o comando primitivo **pinte**. Por exemplo:

```
rg  
repita 4 [pf 40 pe 90]  
pe 45 un pf 5 ul mudel 6  
pinte
```

O quadrado será pintado de vermelho.

Sempre que quisermos pintar uma figura, devemos colocar a tartaruga dentro da área limitada pela figura. Em seguida colo-  
camos o lápis da cor que escolhermos e só então damos o co-  
mando **pinte**. O lápis não pode estar situado sobre qualquer  
linha.

Além de pintar, o HOT-LOGO permite carimbar um determi-  
nado espaço, com figuras da tartaruga em uso e com a cor do  
lápis em questão. Experimente a seqüência:

```
rg mudcf 1 mudel 6  
repita 6 [pf 50 pd 60]  
pd 30 un pf 10 ul mudel 12  
carimabetudo
```

## EDITANDO FIGURAS

O HOT-LOGO nos permite trabalhar com 60 figuras dife-  
rentes, a título de "tartaruga", numeradas de 0 a 59. Veja a  
tabela 3.2.

A tartaruga que utilizamos quando entramos no HOT-LOGO  
é a figura 36. Para "chamarmos" outra figura devemos usar o  
comando primitivo **mudefig** seguido do número da figura dese-  
jada. Por exemplo, tente:

```
mudefig 1  
mudefig 7
```

É possível também alterar uma figura ou criar uma nova.  
Para isso devemos utilizar o editor de figuras.

Com a tela gráfica restaurada (**rg**) e sem tartaruga (**dt**)  
proceda da seguinte maneira para redefinir a figura 10 por  
exemplo:

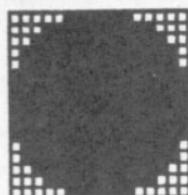
```
edfig 10
```

Na tela aparecerá um grande quadrado em cujo vértice su-  
perior esquerdo localiza-se o cursor piscando.

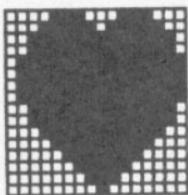
Experimente pressionar as teclas de seta e observe como o cursor se movimenta.

Pressione agora a barra de espaço e uma seta qualquer. Note que o lugar que o cursor estava ocupando ficará em branco. Isto significa que aquele ponto estará apagado na figura 10 que estamos criando.

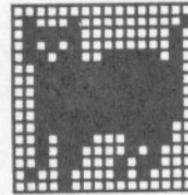
Tabela 3.2 — Figuras do HOT-LOGO



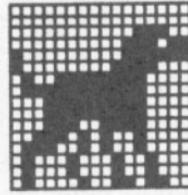
0 - Círculo



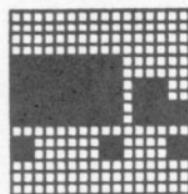
1 - Coração



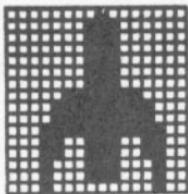
2 - Gato



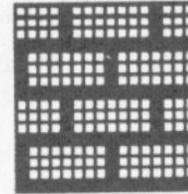
3 - Cachorro



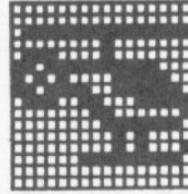
4 - Caminhão



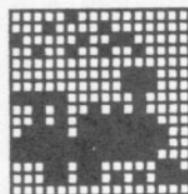
5 - Foguete



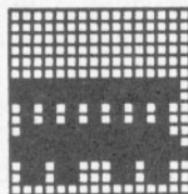
6 - Tijolos



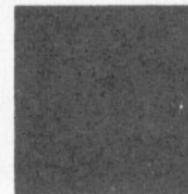
7 - Helicópt.



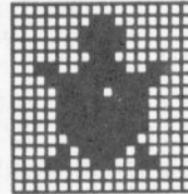
8 - Locomotiva



9 - Vagão



10 a 35:  
Quadrado cheio

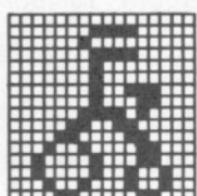


36 a 59:  
Tartarugas

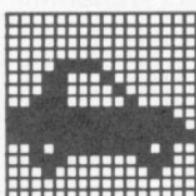
Para treinar um pouco tente fazer a ilustração (a) da figura 3.6.

Figura 3.6

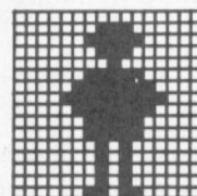
(a)



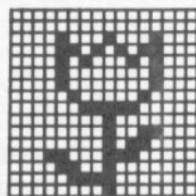
(b)



(c)



(d)



Podemos ainda apagar toda a figura 10 pressionando as teclas CTRL e K simultaneamente.

Quando você terminar sua "obra de arte", saia do modo de edição de figuras apertando a tecla ESC.

Para você visualizar a figura criada comande **at** e em seguida **mudefig 10**.

Para apurar seu senso artístico tente também fazer as ilustrações (b), (c) e (d) da figura 3.6. Para isso, siga os seguintes passos:

```
rg dt edfig 11
```

Copie a ilustração B e tecle ESC.

```
edfig 12
```

Copie a ilustração C e tecle ESC.

```
edfig 13
```

Copie a ilustração D e tecle ESC.

Para "chamar" as figuras criadas, comande:

```
at mudefig 11  
mudefig 12  
mudefig 13
```

## OBJETOS HOT-LOGO

O HOT-LOGO é uma linguagem fácil, simples e interessante para a exploração de jogos, gráficos e geometria. Mas seu poder não pára aí. Como no HOT BASIC, o HOT-LOGO manipula números e cadeias de caracteres através de palavras e de listas compostas de palavras. Números, palavras e listas são chamados de OBJETOS HOT-LOGO.

Em HOT-LOGO uma palavra é uma seqüência de caracteres sem espaços. Uma lista é uma seqüência de palavras ou de outras listas.

Enquanto no HOT BASIC uma cadeia de caracteres deve ser manipulada caracter por caracter, no HOT-LOGO pode ser manipulada como um todo, ou palavra por palavra, ou ainda caracter por caracter.

Como já vimos, o HOT-LOGO superpõe duas telas, a de texto e a gráfica. Para manipular os OBJETOS HOT-LOGO usamos apenas a tela de texto e para tanto tiramos a tartaruga (**dt**).

Para escrever na tela quaisquer OBJETOS HOT-LOGO, usamos o comando primitivo **escreva (esc)**. Digite, por exemplo:

**esc "A1ô**

O resultado aparece na linha seguinte:

**A1ô**

Repare que as aspas são obrigatórias antes da palavra para que o HOT-LOGO não as confunda com nome de procedimentos.

Quando usamos números, as aspas são desnecessárias, pois HOT-LOGO não aceita procedimentos com números como nome. Por exemplo, digite:

**esc 45**

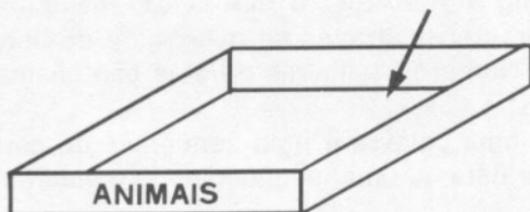
Vamos agora, por exemplo, trabalhar com uma lista de animais. Para isso, digite:

```
atr "animais [gato cachorro ->  
pato sapo]
```

O comando primitivo **atribua (atr)** rotula com um nome um espaço na memória do HOT-LOGO. Com este comando, acabamos de reservar uma "gaveta" de nome animais, onde foi armazenada a linha [gato cachorro pato sapo]. Veja a figura 3.7.

Figura 3.7 — "Gaveta".

[gato cachorro pato sapo]



Para vermos, na tela, o conteúdo de tal gaveta pedimos ao HOT-LOGO:

**esc :animais**

O resultado é:

**gato cachorro pato sapo**

Note o uso de dois pontos (:) pois se comandássemos ao HOT-LOGO esc "animais", teríamos como resultado na tela a palavra animais e não o conteúdo da gaveta chamada animais.

Podemos também concatenar palavras com o conteúdo de uma lista. Comande, por exemplo:

```
esc "animais"\= *animais
```

O resultado é:

```
animais=
gato cachorro pato sao
```

O uso da barra invertida (\) permite tratar o sinal de igual (=) como um caracter normal e não como um comparador.

Para obter a palavra animais e a saída da lista animais numa mesma linha, utilize parênteses:

```
(esc "animais"\= *animais)
```

O resultado é:

```
animais= gato cachorro pato ->
sao
```

É importante salientar que **atribua** permite armazenar qualquer OBJETO HOT-LOGO.

## MANIPULANDO OS OBJETOS HOT-LOGO

Os comandos primitivos **primeiro** (pri), **último** (ult), **sempre-mismo** (sp), **semúltimo** (su), **elemento** e **num.elem** (nel) nos dão inúmeras possibilidades de manipulação dos objetos. São válidos tanto para listas como para palavras.

Vejamos alguns exemplos utilizando como entrada o conteúdo da "gaveta" de nome animais atribuída anteriormente.

```
esc pri *animais
```

O resultado corresponde ao primeiro elemento da lista animais:

```
gato
```

Agora digite:

```
esc ult :animais
```

A resposta é o último elemento da lista:

```
sapo
```

Tente agora:

```
esc elemento 2 :animais
```

O resultado deve ser o elemento número 2 da lista animais.

```
cachorro
```

Comande agora:

```
esc num.elem :animais
```

A resposta deve ser 4, pois trata-se do número de elementos existentes na lista animais.

Vamos agora atribuir e rotular uma outra gaveta na memória do HOT-LOGO:

```
atr "A" "LOGOMANIA
```

Armazenamos na gaveta A a palavra LOGOMANIA. Note que o HOT-LOGO aceita letras maiúsculas como nome de variáveis apesar de não aceitá-las em seus comandos.

Se quisermos ver na tela o último elemento de A faremos:

```
esc ult :A
```

Se quisermos o terceiro elemento de A:

```
esc pri sp sp :A  
ou  
esc elemento 3 :A
```

Imagine agora se pudéssemos digitar um comando e obter na tela as seguintes saídas:

LOGOMANIA	A
LOGOMANI	IA
LOGAMAN	NIA
LOGAMA	ANIA
LOGOM	OU MANIA
LOGO	OMANIA
LOG	GOMANIA
LO	OGOMANIA
L	LOGOMANIA

Para se obter esses resultados, o HOT-LOGO usa dois procedimentos muito simples. Vamos batizá-los de **triangular1** e **triangular2**.

Vamos ensinar o procedimento **triangular1**:

```

ap triangular1 :nome
se :nome = " [pare]
esc :nome
triangular1 su :nome
fim

```

Após pressionar ESC para sair do modo de edição e limpar a tela com att digite:

```

triangular1 "LOGOMANIA
ou
triangular1 :A

```

Veja o resultado.

Teste esse procedimento também com outras palavras. Por exemplo:

```

triangular1 "COMPUTADOR.HOTB->
IT

```

Para obter o segundo resultado ensine **triangular2**:

```

ap triangular2 :nome
se :nome = " [pare]
triangular2 sp :nome
esc :nome
fim

```

Para visualizar o segundo resultado proceda da mesma maneira que a executada em **triangular1**.

Note que com pequenas mudanças no procedimento obtemos resultados muito diferentes.

Vejamos um exemplo simples usando procedimento de listas (lembre-se que listas devem vir sempre entre colchetes).

```
ap sumindo :frase
se :frase =[] [pare]
esc :frase
sumindo su :frase
fim
```

Digite:

```
sumindo [EU VOU SUMIR AGORA] ->
```

Como resultado você deve obter:

```
EU VOU SUMIR AGORA
EU VOU SUMIR
EU VOU
EU
```

## OS OBSERVADORES

O observador é um recurso muito poderoso do HOT-LOGO. Permitindo um modo de programação que funciona independente de todo o procedimento em execução. Ele fica atento a determinados acontecimentos e, assim que os detecta, executa uma lista própria de instruções.

Temos dois comandos primitivos que produzem observadores: **em.colisão** e **quando**.

Os observadores do comando **em.colisão**, vigiam colisão entre tartarugas e os do comando **quando** observam a produção das ocorrências de acordo com a tabela 3.3.

Tabela 3.3

NÚMERO	OCORRÊNCIA
0	aproximadamente uma vez por segundo
1	pressão de uma tecla
2	pressão do botão do joystick 1
3	pressão do botão do joystick 2
4	movimento do joystick 1
5	movimento do joystick 2

Para desativar os comandos **em.colisão** e **quando** usamos **elobs** (elimine observadores) ou **rg** (restaure gráficos).

Para exemplificar, vamos definir um procedimento chamado **demobs**. Sua função será vigiar o teclado. Toda vez que uma

tecla for pressionada, será desenhado um sol e com o comando **toque** será produzido um som seqüencial.

Dessa forma, digite:

```
ap demobs
rg att mudectf i
quando i [sol]
fim

ap sol
tat
atr "l 1
atr "f 60
repita 72 Eraio :l :f atr "l->
:l+i atr "f :f+450
fim

ap raio :l :f
un
pf 20
mudecl :l
ul
pf 50
toque i :f 10 10
un pt 70
pd 5 ul
fim
```

Comande

#### demobs

e veja o efeito produzido.

Para eliminar tal efeito, comande **elobs** ou **rg** mesmo que você não possa visualizar o que teclou.

O procedimento **xoke**, definido a seguir, observa a colisão entre duas tartarugas e quando isso acontece, é escrita a palavra HOT-LOGO:

```
ap xoke
rg
att
mudecf i
atat @ pd 90
mudect 10
atat i pe 90
mudect 13
atat [0 i] at un pf 10
mudevel 20
em.colisão @ i [at mudcurso->
r [10 123 esc EHOT LOGO]]]
fim
```

Note o uso do comando **mudcursor** que coloca o cursor na linha e coluna especificadas.

Para desativar o comando **em.colisão** digite **elobs** ou **rg**.

## OS OPERADORES MATEMÁTICOS

Os operadores matemáticos básicos do HOT-LOGO são: **\***, **+**, **/**, e **-**. Com eles podemos utilizar o computador como uma calculadora.

Vejamos como usá-los na tabela 3.4.

Tabela 3.4 — Operadores Matemáticos

OPERADOR	FUNÇÃO	EXEMPLO	RESPOSTA
*	Produto	esc 30*5	150
+	Adição	esc 2+5	7
/	Divisão	esc 25/5	5
-	Subtração	esc 25-10	15

Os resultados das operações também podem ser usados como entrada para outros comandos. Por exemplo:

```
pt 30 * 2  
pt rq 2500
```

O operador **rq** produz como resultado a raiz quadrada do número indicado.

```
pd (sen 30) * 100 + 40
```

O operador **sen** produz como resultado o seno do número indicado tomado como valor do ângulo, em graus.

Existem ainda outros operadores matemáticos tais como **arctan**, **quociente**, **soma**, etc. Para tanto consulte a listagem dos comandos no final desse capítulo.

Além dos matemáticos, dispomos no HOT-LOGO dos operadores lógicos que nos permitem comparar números ou expressões. Eles são apresentados na tabela 3.5.

Observações:

1 — O comparador **=** pode ser usado com qualquer OBJETO HOT-LOGO (números, palavras e listas).

2 — Os comparadores **>**, **<** e **=**, normalmente são utilizados dentro de programas juntamente com o operador condicional **se**, como já vimos em exemplos anteriores.

3 — Os operadores matemáticos e lógicos são símbolos reservados pelo HOT-LOGO, portanto não podem ser utilizados como caracteres comuns.

Por exemplo, digite:

`esc "*`

e obterá como resposta a seguinte mensagem:

`Não há entradas suficientes ->`  
`para *`

Agora digite:

`esc "\*"`

A resposta agora será:

`*`

Tabela 3.5 — Operadores Lógicos

OPERADOR	EXEMPLOS	RESPOSTA	FUNÇÃO
=	<code>esc 10=3+7</code>	verd	Indicar verd se a expressão for verdadeira e falso caso contrário.
>	<code>esc 10&gt;5</code>	falso	
<	<code>esc 10&lt;5</code>	vera	

Conforme já vimos, a barra invertida (`\`) permite usar os caracteres reservados como se fossem caracteres normais.

## GRAVAÇÃO E CARREGAMENTO

Todos os procedimentos, variáveis e figuras que definimos no HOT-LOGO, são guardados em um local denominado espaço de trabalho (memória livre do computador).

Quando desligamos o computador todo o conteúdo do espaço de trabalho é perdido e se você quiser preservá-lo, ele deve ser armazenado em disco ou fita.

Antes, porém, de gravarmos o espaço de trabalho vamos ver o que nele se encontra. Comande:

`motudo`

O HOT-LOGO mostrará os procedimentos e variáveis armazenados em sua memória.

Se não quisermos gravar todos eles, devemos eliminar do espaço de trabalho os que não nos interessam. Para tanto, usamos os comandos **eln** ou **elns** para eliminar variáveis e **el** ou **elps** para eliminar procedimentos.

Agora, tendo no espaço de trabalho só o que precisamos, podemos gravar da seguinte maneira:

```
gravetudo "NOME  
ou  
gravetudoc "NOME
```

O primeiro comando grava em disco e o segundo em fita cassete.

O nome dado para o arquivo é aleatório, porém sempre deve vir precedido de aspas ("").

Para gravar figuras, o HOT-LOGO utiliza números que representam a configuração dessa figura. Assim, antes de gravar uma figura, devemos armazená-la em uma variável.

Por exemplo, se tivermos editado um robô na figura 10, procedemos da seguinte maneira:

```
atr "robô listafig 10
```

Feito isso, o robô está armazenado na memória do computador. Para gravá-lo, basta usar **gravetudo** (ou **gravetudoc**) sucedido pelo nome que quiser dar ao seu arquivo. Não se esqueça das aspas.

Para carregar um arquivo do disco ou fita para a memória do computador, usamos, respectivamente, os comandos:

```
carregue "NOME  
ou  
carreguec "NOME
```

## DICIONÁRIO DO HOT-LOGO

### ENTRADAS

A seguir são relacionadas várias palavras, simples ou compostas, sem qualquer espaço entre seus caracteres. Nos diversos comandos e operações do HOT-LOGO cada uma dessas palavras representa uma entrada necessária à sua sintaxe.

CANAL : um inteiro de 0 a 2.

CARACTER : uma letra do alfabeto. Um número ou um sinal de pontuação.

COLUNA : um número inteiro de 0 a 28.

CORNUMERO : um número inteiro de 0 a 15. O HOT-LOGO aceita números maiores, dividindo-os por 16 e utilizando o resto como número de entrada. Exemplo : se o número for 16 o resto será 0 e então este será o número utilizado.

DEC : um número decimal inteiro de 0 a 255.

ENDEREÇO : um número inteiro de 0 a 65535.

FIGURALISTA : lista de 32 números que representam a forma de uma figura de tartaruga.

FREQUENCIA : um número de 40 a 4095.

JOYSTICKNUMERO : o inteiro 1 ou 2.

LINHA : um número inteiro de 0 a 23.

LISTA : informações contidas entre colchetes.

LISTAINSTRUÇÕES : lista de procedimentos executáveis pelo HOT-LOGO, entre colchetes.

LISTANOMES : lista de palavras que nomeiam procedimentos, variáveis ou listas de propriedades. Deve estar entre colchetes.

LISTANUMEROTARTARUGA : lista de números inteiros de 0 a 29, entre colchetes.

NOME : palavra que denomina um procedimento, uma variável ou uma lista de propriedades.

NOMEARQUIVO : palavra única que denomina um arquivo. A palavra pode ter no máximo 8 caracteres para disco e 6 para fita-cassete. Não pode incluir pontos ou vírgulas e deve ser precedida de aspas ("").

NOVONOME : palavra utilizada para denominar um novo procedimento.

NUMERO : um número qualquer.

NUMEROFIGURA ou NUMERONOVAFIGURA : um inteiro de 0 a 59.

OBJETO : pode ser uma palavra, uma lista ou um número.

OCORRENCIANUMERO : um inteiro de 0 a 5.

PALAVRA : uma seqüência de caracteres sem espaços.

PORTA : um número de 0 a 255.

PRED : predicado, é uma operação que produz como saída a palavra "falso" ou "verd".

**PROP** : palavra que denomina uma propriedade.

**TARTARUGANUMERO** : um inteiro de 0 a 29.

**TEMPO** : um inteiro que varia de 0 a 255 (60 equivale a 1 segundo).

**SOM** : um inteiro de 0 a 31.

**VOLUME** : um inteiro de 0 a 15.

**X ou Y** : números que representam as coordenadas x ou y.

## COMANDOS PRIMITIVOS E OPERAÇÕES

**algum PRED1 PRED2**

Indica "verd" se alguma das suas entradas for verdadeira e "falso" se todas as entradas forem falsas. Exemplo:

aprenda teste :x

se algum :x>3 :x>7 [esc [maior que 3]] [esc [menor que 3]]

**apaguedesenho**

(apaguedesenho = ad)

Apaga a tela gráfica sem alterar a posição da tartaruga.

**apaguetexto**

(apaguetexto = att)

Apaga todo o texto da tela.

**apareçatat**

(apareçatat = at)

Torna visível a tartaruga ou as tartarugas que estão sendo usadas.

**aprenda**

(aprenda = ap)

Permite que ensinemos novos comandos ao HOT-LOGO ativando o modo de edição. Para sair do editor pressione ESC. Exemplo:

ap quadrado

**arctan NUMERO**

Significa arco tangente, produz um valor em graus (-90 a 90) da tangente do número. Exemplo:

esc arctan 20

**arquivos**

Lista todos os nomes contidos no arquivo em disco. Equiva-

le a mostrar o índice dos arquivos ou diretório. Exemplo:  
arquivos

### **arredonde NUMERO**

Arredonda o valor de um número para inteiro. Exemplo:  
esc arredonde 2,7

### **ascii CARACTER**

Produz o número correspondente ao código ASCII do caractere em questão. Exemplo:

esc ascii "c

### **atençãotat NUMEROTARTARUGA**

### **atençãotat LISTANUMEROTARTARUGA** (atençãotat = atat)

Ativa a tartaruga ou tartarugas que obedecerão aos comandos subsequentes. Exemplo:

atat 0 pf 50 atat 1 at pt 50 atat 2 at pd 90  
atat [0 1 2] repita 4 [pf 20 pd 90]

### **atribua NOME OBJETO**

(atribua = atr)

Cria uma gaveta na memória do computador que armazenará objetos HOT-LOGO. Exemplo:

atr "cor [amarelo azul branco]  
atr "A 10  
atr "nome "Maria

### **aumentelimit**

(aumentelimit = al)

Estabelece uma modalidade gráfica em que a tela constitue uma janela muito maior, não permitindo que visualizemos a tartaruga fora dos limites da mesma.

### **cada LISTAINSTRUÇÕES**

Permite que cada tartaruga ativada execute seqüencialmente a lista de instruções. Exemplo:

pe 90 atat 1 pd 90 dt  
atat [0 1] cada [repita 360 [pf 0,5 pd 1]]

### **caractere NUMERO**

(caractere = car)

Produz como saída o caracter, cujo código ASCII se especi-

fica em NUMERO. Exemplo:

esc car 65

### **car entrada**

(car.entrada = care)

Interrompe a execução do procedimento, se for o caso e espera uma entrada via teclado (1 caracter). Exemplo:

esc care (e digite qualquer tecla)

(esc [eu pressionei] care)

### **carimbe**

Carimba na tela uma cópia da figura da tartaruga ou tartarugas ativadas na posição que elas se encontrarem. A figura carimbada terá a cor do lápis que estiver sendo usada. Exemplo:

carimbe pf 50

repita 18 [carimbe pf 20 pd 20]

### **carimbetudo**

Carimba a figura da tartaruga ativada com a cor do lápis por toda a tela. Se quisermos carimbar a figura da tartaruga dentro de uma região limitada, basta colocá-la nesta região e não deixá-la sobre o limite. Exemplo: se quisermos um quadrado carimbado com a figura 1.

repita 4 [pf 80 pd 90]

pd 45 un pf 10 ul mudefig1 carimbetudo

### **carregue NOMEARQUIVO**

Carrega o arquivo NOMEARQUIVO do disco para a área de trabalho (memória do computador). Exemplo:

carregue "trabalho"

### **carreguec NOMEARQUIVO**

Significa carregue cassete. Carrega o arquivo NOMEARQUIVO da fita cassete para a área de trabalho. Exemplo:

carreguec "trab"

Observação: É imprescindível que a tecla PLAY do gravador esteja pressionada e que o arquivo NOMEARQUIVO exista, pois senão o HOT-LOGO ficará procurando-o indefinidamente.

### **carreguedes NOMEARQUIVO**

Significa carregue desenho. Carrega o desenho NOMEAR-

QUIVO do disco para a área de trabalho. Exemplo:  
carreguedes "carro

**coloque** OBJETO NOME (coloque = col)

Cria uma variável análoga ao atribua porém, com as entradas invertidas. Exemplo:

```
col 10 "A  
col "Maria "nome  
col [ a b c ] "letras
```

**coloqueprop** NOME OBJETO

Cria uma gaveta com o nome da propriedade que armazena um valor. Exemplo:

```
coloqueprop "zerar "con 0  
ap contar :con  
se :con > 10 [esc :con atr "con prop "zerar "con esc :con]  
[mo :con contar :con + 1]  
fim
```

**comimpressora**

Ativa o canal de saída da impressora. Use-o apenas se você tiver uma.

**congele**

Suspender a velocidade da tartaruga ou tartarugas ativadas até que se descongele. Exemplo:

```
mudevel 20 congele espere 90 descongele
```

**conteúdo** NOME

Indica o conteúdo da variável NOME. Exemplo:

```
atr "forma "triângulo  
esc conteúdo "forma  
triângulo
```

**coorx**

Significa coordenada x. Indica a coordenada x da tartaruga em uso. Exemplo:

```
pd 90 pf 70 esc coorx
```

**coory**

Significa coordenada y. Exemplo:

```
mudey (coory + 30)
```

**copiafig** NUMEROFIGURA NUMERONOVAFIGURA

Significa copia figura. Copia a figura especificada em NUMEROFIGURA para ser a figura de NUMERONOVAFIGURA. Exemplo:

copiafig 1 15

Observação: agora a figura 15 será também um coração.

**copie** NOME NOVONOME

Faz uma cópia de modo de edição do procedimento NOME com NOVONOME. Exemplo:

copie "quadrado" "quadrilátero"

Observação: o procedimento quadrado deve ter sido definido anteriormente.

**cordebaixo**

(cordebaixo = cb)

Indica o número da cor existente sob o lápis da tartaruga. Exemplo:

esc cb

**cordefundo**

(cordefundo = cf)

Indica cor do fundo. Exemplo:

esc cf

**cordolápis**

(cordolápis = cl)

Indica a cor do lápis que a tartaruga está usando. Exemplo:

esc cl

**cortat**

Indica a cor da tartaruga ativada. Exemplo:

esc cortat

**cos** NUMERO

Significa cosseno. Indica o cosseno de NUMERO em graus.

Exemplo:

esc cos 60

**criafigl** NUMEROFIGURA FIGURALISTA

Significa cria figura lista. Concede à NUMEROFIGURA, a

forma definida em FIGURALISTA. Veja explicações de listafig.

Exemplo:

criafgl 10 listafig 3

Observação: fará com que a figura 10 tenha a mesma forma que a figura 3.

#### **cursor**

Indica a posição do cursor na forma de uma lista [x y].

Exemplo:

esc cursor

#### **defina NOME LISTA**

Define um procedimento sem entrar no modo de edição.

Exemplo:

defina "circulo [] [repita 360 [pf 0,5 pd 1]]]

círculo

Observe que o HOT-LOGO exige que a primeira instrução da lista seja o nome da entrada do procedimento. No caso, como círculo não exige entrada, a primeira instrução é uma lista vazia.

#### **desapareçatat**

(desapareçatat = dt)

Torna invisível a tartaruga ou tartarugas ativadas.

#### **descongele**

Desativa o comando congele, fazendo com que as tartarugas voltem a ter a velocidade definida no comando mudevel.

#### **diferença NUMERO1 NUMERO2**

Indica o resultado da subtração dos números indicados.

Exemplo:

esc diferença 50 10

esc -(diferença 10 50)

#### **direção**

(direção = dç)

Indica em graus e no sentido horário a orientação da tartaruga em relação à sua direção original que é a norte. Exemplo:

dç

mudedç 50+dç

#### **direçãopara [X Y]**

Indica em graus o giro a ser dado pela tartaruga para apontar

o ponto [X Y] a partir do norte. Exemplo:  
esc direçãopara [20 20]

### **distância [X Y]**

Indica a distância entre a tartaruga ativada e a posição [X Y]. Exemplo:  
esc distância [25 30]

### **e PRED1 PRED2**

Indica "verd" sempre que todas as entradas forem verdadeiras e "falso" quando uma delas for falsa. Exemplo:  
ap inter :x  
se e :x > 10 :x < 20 [(esc :x [pertence ao intervalo])  
pare]  
(esc :x [não pertence ao intervalo])  
fim

### **edfig NUMEROFIGURA**

Significa edite a figura. Coloca a figura NUMEROFIGURA no modo de edição de figuras. Exemplo:

edfig 10

### **edite NOME ou LISTANOMES** (edite = ed)

Entra no modo de edição de procedimentos, idêntico ao do comando aprenda só que admite a edição de um ou mais procedimentos. Exemplo:

ed "desenho1  
ed [desenho1 desenho2 desenho3]

### **edns**

Significa edite nomes. Coloca no modo de edição todas as variáveis já criadas.

### **elemento NUMERO OBJETO**

Indica o elemento que está na posição NUMERO em OBJETO.  
Exemplos:

esc elemento 3 "Maria"

A resposta deve ser r, pois se trata do elemento número 3 de Maria.

esc elemento 2 [Ana Maria da Silva]

A resposta deve ser Maria, pois se trata do segundo elemento da lista.

**elimine** NOME ou LISTANOMES (elimine = el)

Elimina da área de trabalho o(s) procedimento(s) especificados. Exemplo:

el "circulo

**eliminearq** NOMEARQUIVO

Significa elimine arquivo. Apaga do disco o NOMEARQUIVO. Exemplo:

eliminearq "arquivo1

**eln** NOME ou LISTANOMES

Significa elimine nome. Apaga da área de trabalho o NOME ou LISTANOMES de variáveis especificadas pelos comandos atribua e coloque. Exemplo:

atr "cor [azul amarelo branco] eln "cor

**elns**

Significa elimine nomes. Apaga da área de trabalho todas as variáveis.

**elps**

Significa elimine procedimentos. Apaga todos os procedimentos da área de trabalho.

**eltudo**

Significa elimine tudo. Apaga da área de trabalho todos os procedimentos, variáveis e propriedades. Deve ser usado com muita atenção.

**em.colisão** TARTARUGA1 TARTARUGA2 LISTAINSTRUÇÕES

Ativa um "observador" que ficou atento a uma colisão entre as duas tartarugas determinadas. Quando elas se chocam executa a LISTAINSTRUÇÕES. Veja explicações de observador. Exemplo:

em.colisão 0 1 [mudevel 50]

**envie** OBJETO

Detém o procedimento em curso e envia o resultado para

o nível superior. Deve-se usar dentro de procedimentos.

Exemplo:

```
ap en  
envie "Maria  
fim  
ap frase  
(esc [eu me chamo] en)  
fim
```

#### **escreva** OBJETO

(escreva = esc)

Imprime o OBJETO designado na tela e coloca o cursor na linha seguinte. Exemplos:

```
esc 1  
esc "Maria  
esc [eu sou uma lista]
```

Observação: Este comando só permite uma entrada porém, podemos ampliá-lo usando parênteses. Exemplo:

```
(esc "eu "sou Maria)  
(esc "olá [eu sou o HOT-LOGO])
```

#### **espere** NUMERO

Provoca uma pausa na operação, cuja duração será NUMERO 60 avos de segundo. Exemplo:

```
espere 120
```

Deve provocar uma pausa de aproximadamente 2 segundos.

#### **falso**

Indica a palavra reservada "falso".

#### **faça** LISTA

Manda executar as instruções que estiverem em LISTA. Exemplo:

```
faça elemento 2 [[pf 50] [pt 50] [pd 90] [pe 90]]
```

#### **fig**

Indica o número da tartaruga em uso. Exemplo:  
mo fig

#### **fim**

É usado para determinar o fim de um procedimento ensinado pelo comando aprenda.

**gravedes NOMEARQUIVO**

Significa grave desenho, isto é, armazena a imagem da tela no arquivo NOMEARQUIVO. Exemplo:

gravedes "dese1"

**gravetudo NOMEARQUIVO**

Grava todos os procedimentos, variáveis, propriedades que estiverem na área de trabalho em um arquivo de disco denominado NOMEARQUIVO. Exemplo:

gravetudo "arquivo1"

**gravetudoc NOMEARQUIVO**

Grava tudo que estiver na área de trabalho em um arquivo de fita cassete chamado NOMEARQUIVO. Exemplo:

gravetudoc "arq1"

Observação: NOMEARQUIVO só pode conter 6 caracteres,. Não aceita vírgulas, pontos ou espaços em branco.

**int NUMERO**

Indica a parte inteira de número. Exemplo:

esc int 3,7

**inverte**

Substitui a figura da tartaruga em uso pelo desenho que estiver sob a mesma. Deve-se tomar muito cuidado para usá-lo pois, se a tartaruga em uso for de número 36 a 59, mudar-se-á a sua configuração e não poderemos mais tê-la de volta.

É aconselhável usar o comando inverte apenas com as figuras de 10 a 35 e quando a área sob a tartaruga não estiver em branco, senão teremos uma tartaruga invisível.

**joy JOYSTICKNUMERO**

Indica um número de 0 a 8, que se refere a posição atual do joystick. Exemplo:

esc joy 1

**junt enofim OBJETO LISTA**

(junt enofim = jf)

Acrescenta um OBJETO no fim da LISTA indicada. Exemplo:

mo jf "seis [quatro cinco]

Deve resultar em [quatro cinco seis].

**junt enó inicio** OBJETO LISTA

(junt enó inicio = ji)

Acrescenta um OBJETO no início da LISTA indicada.

Exemplo:

mo ji [um] [dois três]

Deve resultar em [[um] [dois três]].

**liberemem**

Significa libere memória. Rearranja a memória do computador, criando mais espaço.

**lin.entrada**

(lin.entrada = line)

Interrompe a execução do procedimento, se for o caso, e espera uma entrada via teclado, devolvendo uma lista depois de pressionada a tecla RETURN. Exemplos:

mo line

esc line

**lista.OBJETO1 OBJETO2**

Indica uma lista formada por suas entradas. Se as entradas forem duas palavras, formará uma lista com as duas porém, se as entradas forem duas listas formará uma terceira lista contendo as duas primeiras. Comando similar a sentença, porém conserva os colchetes das entradas. Exemplos:

esc (lista "eu "sou "uma "lista)

esc linha [eu sou] [uma lista]

Observação: usando o parênteses podemos utilizar lista com mais de duas entradas.

**listafig NUMEROFIGURA**

Produz uma lista de 32 elementos que representa a configuração da tartaruga NUMEROFIGURA. Utilizado na criação de figuras por lista normalmente com o crafogl. Exemplo:

atribua "coração listafig 1

**memlivre**

Significa memória livre. Indica a área de trabalho disponível.

**mons**

Significa mostre nomes. Imprime na tela o nome e o conteúdo de todas as variáveis armazenadas na área de trabalho.

### **moobs**

Significa mostra observadores. Mostra todos os "observadores" definidos pelos comandos em.colisão e quando. Veja explicação de observadores.

### **mop NOME LISTANOMES**

Significa mostra procedimento. Mostra o procedimento NOME ou os procedimentos especificados em LISTANOMES que estiverem na área de trabalho. Exemplos:

mop "círculo"

mop [círculo quadrado retângulo]

### **moprop**

Significa mostra propriedades. Mostra todas as propriedades incluídas na área de trabalho.

### **mops**

Mostra procedimentos. Exibe na tela todos os procedimentos incluídos na área de trabalho.

### **mostra OBJETO**

(mostra = mo)

Mostra na tela o OBJETO especificado. Se for uma lista conserva os colchetes. O cursor é colocado na linha seguinte. Exemplo:

atr "cor [azul amarelo verde]

mo : cor

Deve resultar em [azul amarelo verde].

### **mostrearq NOMEARQUIVO**

Significa mostre arquivo. Mostra o conteúdo do arquivo de nome especificado por NOMEARQUIVO que esteja armazenado em disco. Exemplo:

mostrearq "meuarq

### **mots**

Significa mostra títulos. Mostra o nome (títulos) de todos os procedimentos incluídos na área de trabalho.

### **motudo**

Significa mostra tudo. Mostra todos os procedimentos, variáveis e propriedades incluídas na área de trabalho.

**mudecf CORNUMERO**

Significa mude cor do fundo. Muda a cor do fundo para a CORNUMERO especificada. Exemplo:

mudecf 1

**mudecl CORNUMERO**

Muda a cor do lápis da tartaruga ou tartarugas ativadas para a cor especificada em CORNUMERO. Exemplo:

mudecl 3

**mudecor CORNUMERO1 CORNUMERO2**

Troca a cor do gráfico desenhado na tela e do texto da CORNUMERO1 para a CORNUMERO2. Exemplo:

mudecor 15 8

**mudect CORNUMERO**

Muda a cor da tartaruga ou tartarugas para a cor especificada em CORNUMERO. Exemplo:

mudect 8

**mudecursor [COLUNA LINHA]**

Muda o cursor para a [COLUNA LINHA] especificada.

Exemplo:

mudecursor [20 10]

**mudedç NUMERO**

Muda a orientação (em relação ao norte) da tartaruga ou tartarugas, colocando-as na direção especificada em NUMERO em graus. Exemplo:

mudedç 90

**mudefig NUMEROFIGURA**

Muda a figura da tartaruga ou tartarugas ativadas para a figura especificada em NUMEROFIGURA. Exemplo:

mudefig 3

**mudepos [X Y]**

Coloca a tartaruga ou tartarugas ativadas na posição [X Y] especificada. Não se esqueça que a tartaruga deixa rastro.

Exemplo:

mudepos [-30 40]

**mudeproporção** NUMERO

Modifica para NUMERO o tamanho do passo da tartaruga no eixo x em relação ao eixo y. O valor habitual de proporção é 1,1. Exemplo:

mudeproporção 0,5

**mudeteto** LINHA

Apaga a tela de texto e fixa o cursor na LINHA determinada, deixando a margem superior livre para gráficos. Exemplo:  
mudeteto 20

**mudevel** NUMERO

Muda a velocidade da tartaruga ou tartarugas ativadas para o número especificado. A velocidade normal da tartaruga é 0 e pode variar de -128 a 128.

**mudevelx** NUMERO

Significa muda a velocidade x. Estabelece a componente x da velocidade da tartaruga ou tartarugas ativadas fixando-a em NUMERO. Exemplo:

mudevelx 20

**mudevely** NUMERO

Significa muda a velocidade y. Estabelece a componente y da velocidade da tartaruga ou tartarugas ativadas fixando-a em NUMERO. Exemplo:

mudevely 10

**mudex** X

Faz com que a tartaruga se desloque para a posição X especificada. Não altera a posição y. Exemplo:

mudex 30

**mudey** Y

Mantem a posição X da tartaruga, fazendo com que ela se desloque para a posição Y especificada. Exemplo.

mudey 70

**num.elem** OBJETO

(num.elem = nel)

Devolve o número do elemento que compõe OBJETO. Se for uma lista, devolve o número de palavras. Se for uma palavra,

devolve o número de caracteres. Exemplos:

esc nel [ABCD]

esc nel "Hot

### **nívelinicial**

Cancela a execução de todos os procedimentos e devolve o controle ao HOT-LOGO. Exemplo:

esc [vou voltar ao Logo] nívelinicial esc "ADEUS

### **não PRED**

Indica "verd" se PRED for falso, ou "falso" se PRED é verdadeiro. Exemplo:

ap émaior :X

se não :X>0 [esc [menor que zero]] [esc [maior que zero]]

fim

é maior 10

Observação: Não entre o valor zero neste exemplo ou ocorrerá um erro.

**palavra PALAVRA1 PALAVRA2** (palavra = pal)

Concatena PALAVRA1 com PALAVRA2 sem deixar espaços.

Exemplo:

esc pal "Mari "ana

esc (palavra "Ma "ri "ana)

Ambos os exemplos devem resultar em Mariana.

### **paracentro**

(paracentro = pc)

Leva a tartaruga ou tartarugas ativadas para o centro da tela deixando-as na sua posição original. Não se esqueça que a tartaruga deixa rastro ao se deslocar.

### **paracentroa**

(paracentroa = pca)

Leva a tartaruga ou tartarugas ativadas para o centro e apaga a tela gráfica.

### **paradireita NUMERO**

(paradireita = pd)

Vira a tartaruga ou tartarugas ativadas para a direita de um número de graus especificado por NUMERO. Exemplo:

pd 90

**paraesquerda** NUMERO (paraesquerda = pe)

Vira a tartaruga ou tartarugas ativadas para a esquerda de um número de graus especificado por NUMERO: Exemplo:  
pe 90

**parafrente** NUMERO (parafrente = pf)

Desloca a tartaruga ou tartarugas ativadas para frente o número de passos especificado por NUMERO. Exemplo:  
pf 100

**paratrás** NUMERO (paratrás = pt)

Desloca a tartaruga ou tartarugas ativadas para trás o número de passos especificado por NUMERO. Exemplo:  
pt 100

**pare**

Interrompe o procedimento ativado. Só se usa dentro de procedimentos.

**pinte**

Preenche a tela, ou parte dela, que esteja limitada com a cor do lápis usado pela tartaruga. Esse comando só tem efeito se a tartaruga não estiver sobre um traço. Exemplo:

mudecl 8 pinte

**ponha** OBJETO

Mostra OBJETO na tela e coloca o cursor na mesma linha. Se usado com mais de uma entrada, concatena sem deixar espaço. Análogo ao comando palavra. Exemplos:

ponha "ola  
(ponha "ola "lá)

**ponhaponto** [X Y] (ponhaponto = pp)

Coloca um ponto na coordenada [X Y]. Exemplo:

mudecf 1  
pp [20 20]

**posição** (posição = pos)

Indica a posição atual da tartaruga ativada na forma [X Y]. Exemplo:

esc posição

**primeiro** OBJETO

(primeiro = pri)

Devolve o primeiro elemento do OBJETO. Exemplos:

esc pri "Logo

esc pri [gato cachorro pato]

**primitivas**

Mostra na tela a lista de palavras primitivas (comandos) reservadas no HOT-LOGO.

**produto** NUMERO1 NUMERO2

Indica o produto de NUMERO1 por NUMERO2. Exemplos:

esc produto 3 5

esc (produto 3 5 10)

**proporção**

Devolve a atual proporção entre o passo vertical e o passo horizontal da tartaruga. Exemplo:

esc proporção

**propriedade** NOME PROP

(propriedade = prop)

Devolve o valor da propriedade NOME. Exemplo:

coloque prop "tartaruga" cor 5

esc prop "tartaruga" cor

**quando** OCORRENCIANUMERO LISTAINSTRUÇÕES

Estabelece um "observador" para que execute a LISTAINSTRUÇÕES quando houver uma OCORRENCIANUMERO. Veja explicações de observador. Exemplo:

quando 0 [pf 5 pd 5]

**quem**

Mostra a lista de tartarugas que estão ativadas no momento. Exemplo:

esc quem

**quociente** NUMERO1 NUMERO 2

Devolve o resultado da divisão de NUMERO1 por NUMERO2. Exemplo:

esc quociente 30 5

**raizq** NUMERO (raizq = rq)

Indica a raiz quadrada de NUMERO. Exemplo:

esc rq 9

**repita** NUMERO LISTAINSTRUÇÕES

Executa a LISTAINSTRUÇÕES o NUMERO de vezes determinado. Exemplo:

repita 4 [esc "HOTBIT"]

**reproduza**

Armazena o número ou seqüência de números sorteados.

Exemplo:

reproduza sorteieaté 10

**resto** NUMERO1 NUMERO 2

Indica o resto da divisão do NUMERO1 pelo NUMERO2.

Exemplo:

esc resto 7 3

**retireprop** NOME PROP

Retira a propriedade PROP e seu valor da lista de propriedade especificada em NOME. Exemplo:

retire prop "zerar" "con

**retiretodasprop**

Apaga todas as listas de propriedades da área de trabalho.

**rg**

Significa recomeça gráficos. Restaura a tela gráfica à sua condição inicial. Apaga os observadores, os desenhos, leva as tartarugas para o centro e devolve a cor branca para textos e desenhos e azul para o fundo.

**sãoiguais** OBJETO1 OBJETO2

Verifica se OBJETO1 e OBJETO2 são iguais devolvendo "verd" ou "falso". Exemplos:

esc sãoiguais "A" "a

esc sãoiguais 2\*3 6

**se** PRED LISTA1 LISTA2

Se PRED for "verd" executa a LISTA1 de instruções senão

executa a LISTA2 ou a linha seguinte do procedimento.

Este comando deve ser escrito na mesma linha e LISTA2 é um parâmetro opcional. Exemplo:

se posição = [0 0] [pf 90] [pt 90]

### **semimpressora**

Desativa o canal de saída da impressora.

### **semprimeiro** OBJETO (semprimeiro = sp)

Retira o primeiro elemento de OBJETO e retorna o que restou. Exemplo:

esc sp "Logo

### **semúltimo** OBJETO (semúltimo = su)

Retira o último elemento de OBJETO e retorna o que restou. Exemplo:

esc su [HOT LOGO]

### **sen** NUMERO

Indica o seno do ângulo especificado em graus por NUMERO. Exemplo:

esc sen 90

### **sentença** OBJETO1 OBJETO2 (sentença = sn)

Concatena suas entradas com espaços formando uma lista única. Retira os colchetes externos de objeto. Análogo ao comando lista. Exemplos:

esc sn "a [b c]

esc (sn "a "b [c [d]])

### **som** TOM VOLUME TEMPO

Emite um ruído com as características TOM, VOLUME e TEMPO. Exemplo:

som 20 7 50

### **soma** NUMERO1 NUMERO2

Indica a soma dos números fornecidos. Exemplo:

esc soma 25 36

esc (soma 20 10 40)

**sorteiate NUMERO**

Gera um número pseudo-aleatório de 0 a 1. Exemplo:  
repita 3 [esc sorteiaté 10]

**sótat TARTARUGANUMERO LISTA ou  
LISTANUMEROTARTARUGA LISTA**

Permite que a TARTARUGANUMERO ou a LISTANUMERO TARTARUGA executem uma LISTA de instruções. Não altera a lista de tartarugas ativadas pelo comando atat (atençäotat)  
Exemplos:

sótat [1 3] [repita 3 [pf 50 pf 120]]  
sótat 2 [pf 50]

**tartaruga**

(tartaruga = tat)

Apaga a tela gráfica e de texto, colocando a tartaruga no seu estado original ou seja, no centro da tela com velocidade 0 e orientada para o norte.

**temcar**

Indica "verd" depois que alguma tecla for pressionada. Em caso contrário indica "falso". Exemplo:  
esc temcar

**temvalor NOME**

Indica "verd" sempre que o nome tenha sido definido como variável. Em caso contrário indica "falso". Exemplo:  
esc temvalor "nome"

**texto NOME**

Indica em forma de lista as instruções do procedimento NOME. Exemplo:  
mo texto "quadrado"

**tirelimite**

(tirelimite = tl)

Quando a tartaruga sai da tela, permite a continuidade do desenho no lado oposto da mesma. Corresponde ao estado normal do HOT-LOGO.

**todas**

Gera uma lista de 0 a 29. É utilizado para ativar todas as

30 tartarugas. Exemplo:  
atat todas pf 30

#### **todasprop NOME**

Indica uma lista de todas as propriedades relacionadas com NOME. Para criar propriedades usa-se o comando coloqueprop.  
esc todasprop "zerar"

#### **toque CANAL FREQUENCIA VOLUME TEMPO**

Emite um som pelo CANAL desejado, na FREQUENCIA, VOLUME e duração (TEMPO) especificados. Exemplo:  
toque 2 500 10 50

#### **troquesinal NUMERO ou NOME**

Indica NUMERO ou conteúdo de NOME com sinal invertido.  
Exemplos:

esc troquesinal 7  
en (soma 7 troquesinal 4)

#### **useborracha**

(useborracha = ub)

Faz com que a tartaruga ou tartarugas ativadas apaguem as linha por onde passarem. Para desativar usa-se ul.

#### **useinversor**

(useinversor = ui)

Permite que a tartaruga ou tartarugas ativadas tracem linhas onde elas não existam ou apague-as se já existirem. Para desativar usa-se ul. Exemplo:

quadrado ui quadrado

Observação: o procedimento quadrado deve ser definido previamente.

#### **uselápis**

(uselápis = ul)

Faz com que a tartaruga use lápis e deixe rastro por onde passar. Corresponde ao estado natural do HOT-LOGO.

#### **usenada**

(usenada = un)

Permite que a tartaruga se locomova sem deixar rastro. Para desativar use ul. Exemplo:

un pf 100

**vel**

Significa velocidade. Indica a velocidade da tartaruga ativada. Exemplo:

esc vel

**velx**

Indica a componente horizontal da velocidade da tartaruga ativada.

**vely**

Indica a componente vertical da velocidade da tartaruga ativada.

**verd**

Indica a palavra "verd".

**versão**

Indica a versão do seu HOT-LOGO.

**último** OBJETO

(último = ult)

Devolve o último elemento de OBJETO. Se o objeto for uma palavra, devolve um caracter, se for uma lista, devolve uma palavra. Exemplo:

esc ult "Logo"

esc ult [HOT LOGO]

**ébotão** JOYSTICKNUMERO

Indica "verd" sempre que o botão do JOYSTICKNUMERO estiver pressionado; caso contrário, produz "falso". Exemplo:

esc ébotão 2

**éjoy** JOYSTICKNUMERO

Indica "verd" sempre que o JOYSTICKNUMERO estiver acionado; caso contrário, produz "falso". Exemplo:

esc joy 2

**élista** OBJETO

Indica "verd" sempre que OBJETO for uma lista; caso contrário, indica "falso". Exemplo:

atr bicho [gato pato]

esc élista :bichos

**énúmero** OBJETO

Indica "verd" sempre que OBJETO for um número; caso contrário, indica "falso". Exemplo:

esc énúmero 10

**épalavra** OBJETO

Indica "verd" sempre que OBJETO for uma palavra; caso contrário, indica "falso". Exemplo:

esc épalavra "Logo"

**éprimitiva** OBJETO

Indica "verd" sempre que OBJETO for uma palavra reservada do HOT-LOGO. Exemplo:

esc éprimitiva "palavra"

**éprocedimento** NOME

Indica "verd" sempre que NOME for um procedimento definido e que esteja na área de trabalho; caso contrário, indica "falso". Exemplo:

esc éprocedimento "9"

**évazia** OBJETO

Indica "verd" sempre que OBJETO seja vazio; caso contrário, indica "falso". Exemplo:

évazia "palavra"

**évisível**

Indica "verd" sempre que a tartaruga ativada for visível; caso contrário, indica "falso". Exemplo:

évisível

## COMANDOS PRIMITIVOS ESPECIAIS

Os comandos primitivos especiais no HOT-LOGO vem precedidos de ponto (.) para evidenciar sua importância e o seu perigo.

Os comandos especiais **.entra** e **.sai** são específicos para ver ou enviar mensagens a qualquer uma das 256 portas de entrada e saída disponíveis.

Os comandos **.chame**, **.deposite** e **.examine** permitem utili-

zar rotinas em linguagem de máquina e ter acesso direto à memória do computador.

A seguir, mostramos a sintaxe e descrição dos comandos especiais.

#### **.entra PORTA**

Produz como saída o número que tenha encontrado na porta especificada pelo número PORTA. Exemplo:

```
esc .entra 168
```

#### **.sai PORTA DEC**

Envia à porta especificada pelo número PORTA o número DEC. Exemplo:

```
ap zumbido
```

```
.sai 170 255 .sai 170 127 zumbido
```

```
fim
```

Execute o procedimento zumbido e adivinhe o resultado.

#### **.chame ENDEREÇO**

Transfere o controle a uma sub-rotina em linguagem de máquina que começa no endereço especificado pelo número ENDEREÇO. Exemplo:

```
.chame 192
```

Deve ser produzido um "beep".

#### **.deposite ENDEREÇO DEC**

Escreve no endereço especificado pelo número ENDEREÇO o número DEC. Exemplo:

```
.deposite 65533 12
```

#### **.examine ENDEREÇO**

Produz como saída o número que se encontra armazenado no endereço especificado pelo número ENDEREÇO. Exemplo:

```
esc .examine 65533
```

### **CARACTERES ESPECIAIS**

Há uma série de caracteres que em HOT-LOGO têm um significado especial. Para entendê-los passaremos a descrever suas funções e sintaxes.

" (aspas)

Colocadas antes de qualquer palavra, indicam que essa palavra é um conjunto de caracteres que deve ser utilizado conforme foi digitado. Não é, por exemplo, um nome de procedimento ou conteúdo de uma variável. Os números não precisam de aspas.  
Exemplo: esc "Maria

: (dois pontos)

Colocados antes de uma palavra, indicam que estamos nos referindo ao conteúdo de uma variável.

Exemplo: atr "A 10

esc :A

[ ] (colchetes)

Os colchetes são usados para definir listas. Sempre que uma entrada for uma lista, deverá vir entre colchetes.

Exemplos: esc [Maria José]

atat [0 1 2 3] repita 4 [pt 10 pd 90]

( ) (parênteses)

São utilizados para modificar a ordem das operações matemáticas. Permitem ainda variar o número de entradas de alguns comandos primitivos e operadores.

Exemplos: esc 2\*(5+2)

(esc "batata "frita "gostosa)

esc (pal "an "da "va)

esc (soma 20 10 15 40)

(esc (pal "Ma "ri "ana) "gastou (produto 25 450)  
"cruzados.)

\ (barra invertida)

Permite tratar como caracter normal os delimitadores (por exemplo um espaço ou um símbolo matemático). Para tanto, devemos incluí-la antes dos símbolos [, (, ], +, -, \*, /, =, <, >, y, \, e ainda antes de um espaço em branco.

Exemplos: esc "\ \\ \ oi

(esc "3\ \*5\ = 3\*5)

esc [\ \ \ \ HOT\ \ -\ \ LOGO]

# INTRODUÇÃO AO HOT BASIC

Para controlar um computador, você deve dar instruções numa linguagem que ele "entenda". O HB-8000 entende, assim como a maioria dos micro computadores pessoais, várias linguagens como COBOL, ASSEMBLY, C, LISP, BASIC, etc. Para que ele entenda uma determinada linguagem, ela deve ser carregada através de um programa específico.

Uma dessas linguagens, porém, já vem "carregada" da fábrica e o computador já a entende assim que é ligado: é o BASIC (Beginner's All Purpose Symbolic Instruction Code).

Esta linguagem apresenta vários dialetos que diferem entre si nas várias máquinas. O HOT BASIC, ou BASIC MSX, desenvolvido pela Microsoft, é constituído por um conjunto de palavras do idioma inglês e determinadas regras de sintaxe com as quais você pode instruir o computador a realizar inúmeras funções.

A finalidade desse capítulo não é explorar a fundo todos os recursos que o poderoso HOT BASIC oferece mas apenas dar a você um impulso inicial que lhe possibilite fazer seus primeiros programas. Caso haja interesse em aprofundar-se, recomendamos que leia o livro "Manual de Basic" editado pela EPCOM.

## O TECLADO

Para comunicação com o computador deve haver um modo de lhe fornecer instruções e outro dele responder.

Geralmente o teclado é o meio utilizado para dar instruções ao computador. E a tela de TV é o que ele utiliza para respondê-las.

Quando você liga o microcomputador são apresentadas algumas mensagens de identificação do sistema (veja as figuras 1.11 e 1.12) e após alguns segundos deve aparecer a palavra "Ok". Essa é a indicação de que o computador está pronto para aceitar os seus comandos. Surge também um quadrado branco logo abaixo da palavra "Ok". Ele é chamado de "cursor" e sua posição na tela indica o local da próxima letra que você for digitar.

Quanto ao teclado, parece o de uma máquina de escrever. No entanto, ele contém algumas teclas especiais que se fazem

necessárias para que haja uma comunicação efetiva entre você e o computador. Estas teclas encontram-se sombreadas na figura 4.1 e suas funções são descritas a seguir.

(1) **SHIFT** : Essa tecla é usada para digitar letras maiúsculas ou para obter os caracteres impressos na parte superior das teclas. Quando pressionada juntamente com a tecla GRAPH (3), obtém-se o grupo de símbolos gráficos mostrados na figura 1.19.

(2) **CAPS** : Cada vez que essa tecla for pressionada, haverá mudança na apresentação das letras: se estiverem sendo mostradas em minúsculas passam a ser maiúsculas e vice-versa. Possui uma lâmpada que, se estiver acesa, indica o modo "maiúscula". Ela não influi sobre os caracteres impressos na parte superior das teclas.

(3) **GRAPH** ou (4) **CODE** : Mantendo-se uma dessas teclas pressionadas ao digitar-se outras, obtemos a apresentação na tela dos caracteres e símbolos gráficos apresentados nas figuras 1.18 e 1.20.

(5) **RETURN** : Essa tecla é uma das mais importantes e mais utilizadas. Deve ser pressionada invariavelmente após a digitação dos dados, comandos ou instruções para que sejam enviados à memória do computador.

(6) **BACKSPACE** : Ao ser pressionada, cancela o caracter situado à esquerda do cursor. Todos os caracteres situados à direita do cursor permanecem inalterados mas são deslocados para a esquerda ao pressionarmos BACKSPACE.

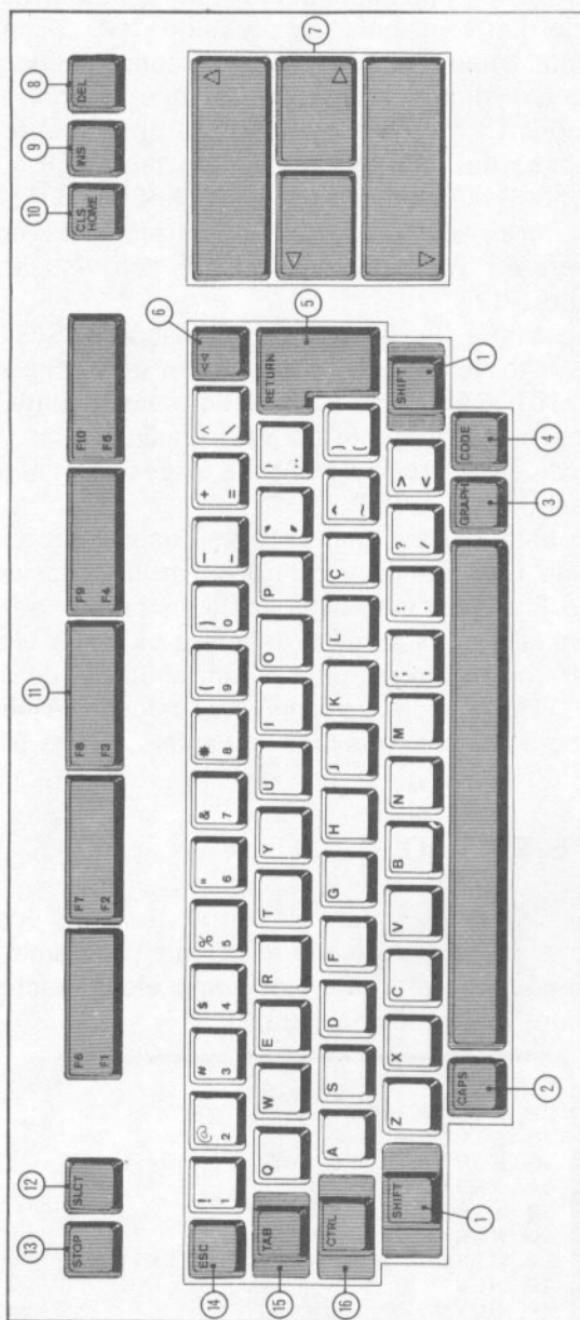
(7) **SETAS** : Essas teclas movem o cursor nas direções e sentidos indicados e não afetam os caracteres já digitados.

(8) **DEL** : Cancela o caracter sobre o qual se encontra o cursor. Todos os caracteres seguintes são deslocados em uma posição à esquerda.

(9) **INS** : Essa tecla deve ser utilizada quando você quiser acrescentar caracteres dentro de uma linha. Mova o cursor para a posição posterior àquela em que você deseja acrescentar algo, pressione INS e digite o texto a ser inserido.

(10) **CLS/HOME** : Pressionando essa tecla, o cursor se posiciona no canto superior esquerdo da tela (posição denominada HOME). Se for pressionada juntamente com a tecla SHIFT, obtém-se a função CLS que significa "Clear Screen", ou seja, "Limpe a Tela". Nesse caso, a tela é limpa e o cursor se desloca para a posição HOME.

Figura 4.1 — Teclas Especiais do HOTBIT.



**(11) F1...F10** : Pressionar uma dessas teclas equivale a introduzir o comando a ela correspondente. Elas existem para reduzir o trabalho de digitação pois, ao pressioná-las, o computador executa o comando nelas programado, evitando-se desta forma, que você tenha que digitar cada caracter que compõe esse comando. As funções F1 até F5 são acionadas diretamente através da simples pressão de cada uma delas. As teclas F6 a F10 são acionadas com a pressão simultânea da tecla SHIFT.

Na parte inferior do vídeo são visualizados os comandos relativos a F1 até F5. Ao pressionar SHIFT, são visualizados os relativos a F6 até F10.

**(12) SLCT** : Não é usada para programação BASIC.

**(13) STOP** : Ao ser pressionada, detém a execução de um programa em HOT BASIC. Pressionando-a novamente, a execução continua a partir do ponto da interrupção.

Para efetivar a interrupção deve-se pressionar simultaneamente as teclas CTRL e STOP.

**(14) ESC** : Essa tecla é muitas vezes usada em aplicações de software. Não tem qualquer efeito se utilizada diretamente.

**(15) TAB** : Pressionando essa tecla, o cursor se desloca para a direita apagando os caracteres sobre os quais ele passar. O cursor só pára ao encontrar uma coluna múltipla de 8.

**(16) CTRL** : Essa tecla, quando pressionada junto com outras, serve para acessar funções especiais, como STOP, por exemplo.

## DIGITAÇÃO E EDIÇÃO

Para que você se familiarize com o teclado, acompanhe passo a passo a digitação de um pequeno programa. Não se preocupe, por enquanto, em entender como ele funciona, pois o que importa agora é aprender a digitar.

Figura 4.2

```
10 COLOR 15,1,1
20 SCREEN 2
30 FOR I=1 TO 20
40 X=RND(1)*256
50 Y=RND(1)*192
60 R=RND(1)*40
70 CIRCLE (X,Y),R,RND(1)*15
80 NEXT I
90 GOTO 90
```

Inicialmente, pressione uma única vez a tecla **CAPS** de modo que sua lâmpada fique acesa. Isso faz com que as letras sejam mostradas em maiúsculo assim como a listagem da figura 4.2.

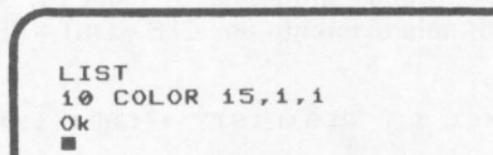
Comece digitando a primeira linha e observe que o cursor vai se movendo para a direita.

Se você errar a digitação de algum caracter, basta apertar **BACKSPACE** e o cursor volta e apaga o caracter errado.

Ao terminar a digitação da primeira linha, pressione a tecla **RETURN**. Dessa forma, o cursor pula para a próxima linha da tela e a linha do programa digitada é armazenada na memória do microcomputador. Para verificar isso, limpe a tela (SHIFT + HOME/CLS) e liste o conteúdo da memória comandando **LIST** seguido da tecla **RETURN**.

A tela deve ficar como o indicado na figura 4.3.

Figura 4.3



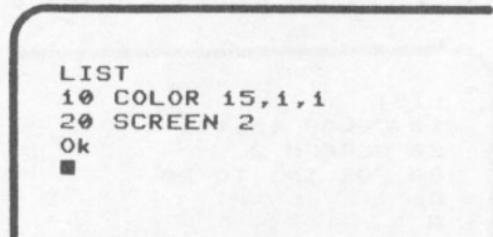
```
LIST
10 COLOR 15,1,1
OK
```

Continuando a digitação do programa, vamos novamente limpar a tela (SHIFT + HOME/CLS) e digitar a linha de número 20.

Não se esqueça de teclar **RETURN** ao chegar ao fim da linha digitada.

Para verificar o conteúdo da memória, limpe novamente a tela (SHIFT + HOME/CLS) e liste o programa com o comando **LIST** (seguido de **RETURN**). A tela deve agora se mostrar conforme a figura 4.4.

Figura 4.4



```
LIST
10 COLOR 15,1,1
20 SCREEN 2
OK
```

Antes de continuar, podemos aprender um pequeno truque que facilita o seu trabalho toda vez que você quiser limpar a tela e a seguir listar o programa. Como essa operação é muito repetitiva, é conveniente atribuí-la a uma tecla de função.

Para programar uma tecla de função, comande:

**KEY n , "(conteúdo do comando)"**

Onde n é o número da tecla que você deseja programar (1 a 10).

Se você quiser que o comando armazenado na tecla seja automaticamente executado sem que a tecla RETURN seja pressionada, então comande:

**KEY n , "(conteúdo do comando)" + CHR\$(13)**

Por exemplo, vamos programar a tecla **F1**, substituindo o comando COLOR nela existente por **CLS+LIST+RETURN**. Assim digite:

**KEY 1 , "CLS:LIST" + CHR\$(13)**

Após pressionar **RETURN**, a listagem das palavras armazenadas nas teclas de funções que aparece na última linha da tela é alterada. Observe que a palavra "color" foi substituída por "CLS:LIST".

Experimente agora apertar a tecla **F1** e veja o resultado. A tela é limpa e o programa é listado sem a necessidade de pressionar RETURN. Isso acontece, pois o código do caracter relativo à tecla RETURN também foi armazenado na programação da tecla **F1** (**CHR\$(13)**).

Digite agora a linha 30 e não esqueça o **RETURN** no final. Aperte a tecla **F1** e a tela que surge é conforme a figura 4.5.

Figura 4.5

```
LIST
10 COLOR 15,1,1
20 SCREEN 2
30 FOR I=1 TO 20
Ok
```

Vamos supor agora que você queira alterar a linha 30 para:

**30 FOR I=1 TO 50**

Para corrigir a instrução que já está armazenada na memória ,temos duas soluções:

a) Redigitar toda a nova linha e, neste caso, ao teclar **RETURN**, a antiga linha 30 é substituída, na memória.

b) Editar a linha que já está na tela. Nesse caso, leve o cursor até o dígito a ser corrigido com o auxílio das teclas do cursor. Tecle o 5 em cima do 2 e pressione **RETURN**. Verifique esse procedimento na figura 4.6.

Figura 4.6

- \* Pressione **█** e **█** até obter:  
**30 FOR I=1 TO 20**
- \* Pressione **█** e obtenha:  
**30 FOR I=1 TO 50**
- \* Tecle **RETURN** e obtenha:  
**30 FOR I=1 TO 50**  
**█**

Aperte agora **F1** e note que a mudança foi executada.  
Agora altere a linha

**30 FOR I=1 TO 50** para **30 FOR I =1 TO 520**

Nesse caso, leve o cursor até o 0 do número 50 e tecle **INS** (INSert). O cursor muda de forma, indicando que o próximo caracter a ser digitado não se sobrepõe ao 0. Digite o 2 que você quer inserir e note que o 0 é deslocado para a direita. Tecle, a seguir, **RETURN**.

Acompanhe o procedimento na figura 4.7.

Figura 4.7

- \* Pressione **█** e **█** até obter:  
**30 FOR I=1 TO 50**
- \* Tecle **INS** e obtenha:  
**30 FOR I=1 TO 50**
- \* Digite **█** e obtenha:  
**30 FOR I=1 TO 520**
- \* Tecle **RETURN** e obtenha:  
**30 FOR I=1 TO 520**  
**█**

Tecle novamente **F1** e perceba a mudança efetuada.

Vamos agora estabelecer a instrução original. Leve o cursor até o caracter 5 e digite **DEL** (Delete) e depois RETURN. Veja a figura 4.8.

Figura 4.8

```
* Pressione [ ] e [ ] até obter:  
30 FOR I=1 TO 20  
* Pressione [DEL] e obtenha:  
30 FOR I=1 TO 20  
* Tecle RETURN e obtenha:  
30 FOR I=i TO 20  
OK
```

Aperte **F1** e continue a digitação do programa com a linha:

```
40 X=RND(1)*256
```

**Não se esqueça do RETURN!**

Olhando para o programa original (figura 4.2), notamos que as linhas 50 e 60 são muito parecidas com a 40. Para poupar trabalho de digitação, podemos usar a linha 40 para gerar as outras duas. Para isso, desloque o cursor para o 4 e digite 5. A seguir, desloque o cursor para o X e digite Y e, finalmente, digite o número 192 sobre o número 256. Ao teclar **RETURN**, essa nova linha é incorporada na memória sem que a 40 seja apagada. Veja a seqüência das operações descritas na figura 4.9.

Figura 4.9

```
* Use a tecla [ ] para obter:  
40 X=RND(1)*256  
* Digite [ ] e obtenha:  
50 X=RND(1)*256  
* Use a tecla [ ] até obter:  
50 [ ]=RND(1)*256  
* Digite [Y] e obtenha:  
50 Y=RND(1)*256  
* Use a tecla [ ] até obter:  
50 Y=RND(1)*256  
* Digite [1][X] e [ ] e obtenha:  
50 Y=RND(1)*192[ ]  
* Tecle RETURN
```

Para se certificar que a nova linha foi armazenada, limpe a tela (SHIFT + HOME/CLS) e comande **LIST 40-50**. Você deve obter na tela a listagem das linhas 40 e 50.

Você pode agora gerar a linha 60, usando a de número 50 conforme a figura 4.10.

Figura 4.10

```
* Use a tecla □ para obter:  
50 Y=RND(1)*192  
* Digite □ e depois □ até obter:  
60 Y=RND(1)*192  
* Digite □ e depois □ até obter:  
60 R=RND(1)*192  
* Digite □,□ e [DEL] para obter:  
60 R=RND(1)*40■  
* Tecla RETURN
```

Aperte agora a tecla F1 e depois digite as linhas 70, 80 e 90, não esquecendo de pressionar **RETURN** após a digitação de cada uma delas.

Não se preocupe com a ordem na qual as linhas são inseridas na memória. Se você introduzir a de número 90 e depois as linhas 70 e 80, o HB-8000 se encarrega de arquivá-las em ordem numérica crescente.

Ao terminar a digitação, aperte a tecla F1 para apagar a tela e listar o programa. A tela deve mostrar-se idêntica à figura 4.2. Confira com cuidado, pois cada vírgula é essencial para que o programa funcione.

Se houver necessidade de efetuar correções, use tudo o que foi aprendido até aqui. E nunca se esqueça de pressionar **RETURN** para inserir uma nova linha na memória.

Outros recursos de edição são mostrados no Apêndice II.

Limpe a tela (SHIFT + HOME/CLS) e execute o programa comandando **RUN** (seguido da tecla RETURN) ou pressionando a tecla F5.

A tela deve se por no modo gráfico e devem ser desenhados vários círculos de tamanhos, cores e em posições diferentes. Para parar o programa e voltar a listá-lo, comande **CTRL+STOP** e depois **F1**.

Quando você desligar o computador, o conteúdo da memória em que são escritos os programas em BASIC (RAM) é apagado. Isso acontece também se for executado o comando **NEW**. Dessa forma, para preservar o programa digitado ele deve ser gravado.

A gravação pode ser feita em fita cassete através do comando **CSAVE**. Posteriormente, a recuperação (leitura) será feita

com o comando **CLOAD**. Para utilizar o gravador para arquivar programas, leia o Apêndice I.

## PROGRAMAÇÃO

O computador executa as instruções a ele fornecidas de duas formas:

I) MODO DIRETO ou IMEDIATO: a instrução é executada assim que for digitada e seguida da pressão da tecla RETURN. Nesse modo, uma linha de instruções não é precedida de um número.

Por exemplo, digite:

```
PRINT"ESSE COMANDO IMPRIME ALGO NA TELA"
```

Pressione RETURN e veja o resultado.

A palavra **PRINT** pode ser substituída por um ponto de interrogação (?). Por exemplo:

```
? CHR$(34);3*7;CHR$(34)
```

deve resultar em:

"21"

Nesse comando executou-se a impressão de aspas ("") com o auxílio da função CHR\$(34). O número que aparece entre parênteses é o código ASCII do caractere desejado (consulte o Apêndice II).

II) MODO INDIRETO ou PROGRAMA: a instrução não é executada imediatamente após sua introdução com a tecla RETURN. Ela é armazenada na memória para posterior execução. Nesse modo, as linhas de instruções devem ser precedidas de um número.

Uma seqüência de instruções contidas em linhas numeradas na ordem em que devem ser executadas é o que chamamos de programa.

Para exemplificar, digite o programa da figura 4.11.

Os números das linhas devem estar compreendidos entre 1 e 65529. Eles normalmente estão em intervalos de 10 para facilitar futuros acréscimos de novas linhas.

Para executar o programa da figura 4.11, pressione a

tecla F5.

Seu funcionamento é simples:

Linha 10: limpa a tela com o comando CLS.

Linha 20: imprime na tela o texto "A RAIZ QUADRADA DE 25 É". O ponto e vírgula (;) colocado no final da linha, instrui o computador a realizar a próxima impressão a partir da posição seguinte à do último caracter impresso.

Linha 30: imprime a raiz quadrada de 25 (SQR(25)="SQuare Root" de 25).

Linha 40: imprime uma linha em branco. Tem o efeito de "pular" uma linha.

Linha 50: imprime a palavra FIM.

Figura 4.11

```
10 CLS
20 PRINT"A RAIZ QUADRADA DE 25 É";
30 PRINT SQR(25)
40 PRINT
50 PRINT"FIM"
```

Se você quiser apagar ou eliminar alguma linha desse programa, isso pode ser feito de dois modos: um é digitar o número da linha a ser apagada e depois teclar RETURN. O outro é usar o comando DELETE, como por exemplo:

DELETE 20: apaga a linha 20.

DELETE 30-50: apaga da linha 30 até a 50.

DELETE —30: apaga desde a primeira linha do programa até a de número 30, inclusive.

Se você quiser acrescentar qualquer linha, basta digitá-la e depois pressionar RETURN. Atribua a ela um número contido no intervalo definido pelos números das duas linhas entre as quais você quer inserí-la. Por exemplo, para incluir uma linha entre as de número 40 e 50 digite:

```
45 PRINT"HB-B000"
```

Há também a possibilidade de colocar mais de uma instrução em uma única linha. Para isso, separe as instruções por dois pontos (: ) e respeite um número máximo de 255 caracteres por linha. Por exemplo:

```
45 PRINT"HB-B000" : PRINT"MEMÓRIA DISPON
IVEL=";FRE(0);"BYTES" = PRINT
```

## AS TELAS DE TEXTO

Existem dois tipos de telas no HB-8000 que permitem escrever textos. A escolha de cada uma delas é feita através do comando **SCREEN**.

Comandando SCREEN 0, é selecionada uma tela de 24 linhas de texto com, no máximo, 40 caracteres cada. Esta é a tela que o computador apresenta assim que é ligado.

Selecionando a outra tela de textos com o comando SCREEN 1, obtém-se uma tela que permite introduzir 24 linhas de texto com no máximo 32 caracteres em cada uma.

O número de caracteres por linha pode ser alterado nas duas telas com o comando **WIDTH**. Seu formato é:

**WIDTH n**

onde, n = número de caracteres por linha. Deve estar compreendido entre 1 e 40 na SCREEN 0 e entre 1 e 32 na SCREEN 1.

Por exemplo, comande:

**SCREEN 0 e depois WIDTH 20**

Note que qualquer texto digitado a partir de agora terá no máximo 20 caracteres por linha. Para retornar ao estado inicial da SCREEN 0 comande WIDTH 39 e da SCREEN 1 execute WIDTH 29.

Uma das diferenças entre a SCREEN 0 e a SCREEN 1 é que na primeira os caracteres gráficos são "truncados" e na segunda eles são impressos "por inteiro". Por exemplo, comande SCREEN 0 e digite as teclas GRAPH e D simultaneamente. A seguir selecione a SCREEN 1 e volte a digitar GRAPH e D. Note a diferença.

Note também que ao selecionar qualquer SCREEN a tela é previamente limpa. Isso ocorre porque é impossível a co-existência de dois tipos diferentes de telas no HOT BASIC.

Manipulando textos, um dos comandos mais utilizados é o **PRINT**. Ele pode vir acompanhado de alguns sinais de pontuação ou de outros comandos que possibilitam organizar a exibição dos textos na tela.

Para exemplificar, digite o programa da figura 4.12.

Figura 4.12

```
10 KEY OFF
20 SCREEN 0 : WIDTH 30
30 PRINT "MICROCOMPUTADOR HOTBIT"
40 PRINT
50 PRINT TAB(5)"HB-8000"TAB(20)"SHARP"
60 LOCATE 2,6
70 PRINT "TEXTO POSICIONADO POR LOCATE"
80 PRINT
90 PRINT "ABC", "123"
100 PRINT "ABC"; "123"
110 PRINT
120 PRINT "TEXTO SEM"
130 PRINT "PONTO E VÍRGULA"
140 PRINT
150 PRINT "TEXTO COM ";
160 PRINT "PONTO E VÍRGULA"
```

A linha 10 do programa utiliza o comando **KEYOFF** que simplesmente "desliga" a visualização na parte inferior da tela dos comandos associados às teclas de função. Para visualizá-los novamente utilize **KEYON**.

A linha 20 seleciona a SCREEN 0 e define uma largura de 30 caracteres para o texto com o comando **WIDTH**.

A linha 30 efetua uma impressão a partir da primeira posição da tela.

A linha 40 tem o efeito de "pular" uma linha de texto na tela.

A linha 50 efetua uma impressão a partir da coluna 5 e outra a partir da coluna 20 contadas desde a margem esquerda da tela. Ela utiliza a função **TAB** (**TABular**), igual ao tabulador de uma máquina de escrever.

A linha 60 utiliza o comando **LOCATE** que tem por função posicionar a próxima impressão. Sua sintaxe é:

### **LOCATE a,b**

onde, a = número da coluna (0 a 39).

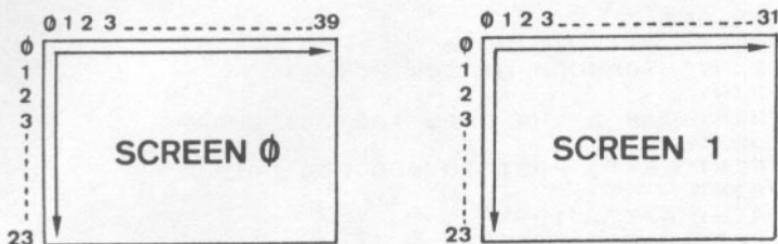
b = número da linha (0 a 23).

As coordenadas das telas de texto são definidas de acordo com a figura 4.13.

As linhas 90 e 100 mostram a diferença entre separar textos por vírgulas ou ponto-e-vírgulas.

As demais linhas do programa são auto-explicativas.

Figura 4.13 — Coordenadas das telas de texto.



## CORES

Quando se liga o HB-8000, a tela apresentada é a SCREEN 0 em fundo azul escuro e caracteres impressos em branco. Essas cores podem ser alteradas com o comando COLOR. Sua sintaxe é a seguinte:

**COLOR x,y,z**

onde, x = código da cor do texto (frente).

y = código da cor da tela (fundo).

z = código da cor das bordas da tela.

Os códigos e as cores correspondentes são mostrados na tabela 4.1.

Tabela 4.1

CÓDIGO DA COR	NOME DA COR
0	incolor
1	preto
2	verde
3	verde claro
4	azul escuro
5	azul claro
6	vermelho escuro
7	azul anil
8	vermelho
9	vermelho claro
10	amarelo ouro
11	amarelo
12	verde escuro
13	roxo
14	cinza
15	branco

Se, por exemplo, você comandar:

**COLOR 1,11**

deve obter caracteres pretos em fundo amarelo.

Experimente agora:

**COLOR 8,8**

O texto desaparece e a tela fica toda vermelha. Se você digitar qualquer coisa, apenas ouvirá o "clic" das teclas mas os caracteres digitados não aparecerão.

Isso ocorre porque você está tentando escrever com tinta vermelha sobre papel vermelho. Para voltar a visualizar os caracteres pressione RETURN duas ou três vezes e a seguir pressione SHIFT + F1 (F6).

O terceiro número do comando COLOR é referente à cor da borda. Ele não foi utilizado até agora pois a borda não aparece em SCREEN 0. Se, por outro lado, você estiver em SCREEN 1, podem ser especificadas as três cores: de frente, do fundo e da borda. Por exemplo, para obter-se texto em cinza sobre tela vermelha de bordas amarelas, comande:

**SCREEN 1 e depois COLOR 14,8,11**

## **CONSTANTES E VARIÁVEIS**

A memória do computador está capacitada a armazenar não somente linhas com instruções mas também outros elementos bastante importantes para a elaboração e execução de programas. Esses elementos são as constantes e as variáveis.

### **CONSTANTES**

As constantes contêm informações que não mudam durante a execução do programa. Elas podem ser de dois tipos: alfanumeréricas ("string") ou numéricas.

As constantes alfanumeréricas podem conter um máximo de 255 caracteres que devem sempre ser colocados entre aspas (""). Por exemplo, "COMPUTADOR" ou "2378". Note que apesar dos caracteres que compõem a constante "2378" serem números, ela não é tratada como um e não poderá ser utilizada diretamente em cálculos.

Uma constante numérica pode ser positiva ou negativa, inteira ou não. No computador, assim como nas máquinas de cal-

cular, a vírgula decimal deve ser substituída por um ponto. Por exemplo, o número 378,75 deve ser escrito como 378.75 (notação anglo-saxônica).

O HOT BASIC reconhece seis tipos de constantes numéricas:

1) **INTEIRA** : número inteiro compreendido entre -32768 e 32767. Exemplos: 279 ou -577.

2) **PONTO FIXO** : número real positivo ou negativo que contém casas decimais. Exemplos: 72.7 ou -23.15.

3) **PONTO FLUTUANTE** : número positivo ou negativo representado na forma exponencial, análoga à notação científica. Uma constante com ponto flutuante é constituída por uma constante inteira ou de ponto fixo (positiva ou negativa) seguida da letra **E** (ou **D**) e o número de posições que o ponto decimal deve andar (se o deslocamento for para a esquerda, esse número é negativo).

Exemplos: -373.97E5 = -37397000 ou  
797.2E-3 = .7972

As constantes de ponto flutuante devem estar compreendidas entre 10 elevado a -64 e 9.99999999999999 vezes 10 elevado a 62.

4) **HEXADECIMAL** : números hexadecimais precedidos por **&H**. Exemplo: &HFB.

5) **OCTAL** : números octais identificados pelo prefixo **&O**. Exemplo: &O37.

6) **BINÁRIA** : números binários precedidos por **&B**. Exemplo: &B10001101.

As constantes numéricas podem também ser declaradas como números de precisão simples (6 algarismos significativos) ou dupla (14 algarismos significativos).

O computador reconhece uma constante como sendo de precisão simples quando, em ponto flutuante, utiliza-se a letra **E**, ou então se ela é seguida por um ponto de exclamação (!). Por exemplo:

378.23E9 ou  
277.23!

A constante é tratada como de precisão dupla se, em ponto flutuante, for utilizada a letra **D** ou se ela for seguida por **#**. Exemplos:

279.2D-3 ou  
44.55 #

Além disso, se nenhuma expoente, ! ou # for atribuído a uma constante ela será tratada com precisão dupla.

## VARIÁVEIS

As variáveis podem ser entendidas como "caixas" que guardam informações. Essas informações podem ser alfanuméricas ou numéricas e podem ser atribuídas diretamente pelo programador ou através de cálculos realizados pelo programa.

Para que o computador saiba localizar as informações, cada "caixa" que as guarda deve ter um nome.

O nome de uma variável pode ter qualquer tamanho. Ele pode conter letras ou números mas o primeiro caracter deve ser sempre uma letra.

Além disso, o nome da variável não pode coincidir com o de qualquer comando, função ou termos do HOT BASIC, tais como OR, PRINT, IF, etc.

Convém notar também que o computador considera apenas os DOIS PRIMEIROS caracteres do nome de uma variável para identificá-la. Isto é, os nomes PREÇO e PRAZO são idênticos para o BASIC-MSX pois ambos começam com PR.

Assim como as constantes, as variáveis podem ser alfanuméricas ou numéricas.

As alfanuméricas devem sempre terminar com \$ e as numéricas podem terminar com % (inteiros), ! (precisão simples) ou # (precisão dupla).

Quando o nome de uma variável não termina com qualquer um dos caracteres mencionados acima (\$, %, ! ou #), o HOT BASIC a considera numérica e de precisão dupla.

Exemplos:

A\$ = nome de variável alfanumérica.

I% = nome de variável numérica inteira.

AZ! = nome de variável numérica de precisão simples.

PI# = nome de variável numérica de precisão dupla.

PI = nome de variável numérica de precisão dupla.

## MANIPULANDO AS VARIÁVEIS

Assim que você liga o computador, todas as "caixas" que podem conter informações (variáveis) estão vazias. Se as variáveis são numéricas, elas contêm inicialmente zeros (0) e caso

sejam alfanuméricas contêm cadeias vazias de caracteres ("").

Para "guardar" alguma informação numa variável, devemos usar o comando LET. Esse comando atribui a uma variável uma informação como se a estivesse guardando na "caixa" correspondente.

Por exemplo, rode o programa da figura 4.14.

Figura 4.14

```
10 SCREEN 0
20 LET A=20
30 LET B=6
40 LET A$="multiplicado por"
50 LET P=A*B
60 LET B$="é igual a"
70 LOCATE 0,10
80 PRINT A;A$;B;B$;P
90 END
```

No programa anterior foram atribuidos os valores 20 e 6 às variáveis A e B, respectivamente. Na linha 50 foi feito o produto dos dois números e o resultado foi "guardado" na variável P.

Se você quiser, não há necessidade de utilizar o comando LET para atribuir algum conteúdo a uma variável. Assim, as linhas 20 a 60 podem ser substituídas pelas que aparecem na figura 4.15.

Figura 4.15

```
20 A=20
30 B=6
40 A$="multiplicado por"
50 P=A*B
60 B$="é igual a"
```

Acrescente agora a linha:

25 A=30

Note que a informação guardada na variável A na linha 20 ( $A = 20$ ) é simplesmente substituída na linha 25 ( $A = 30$ ).

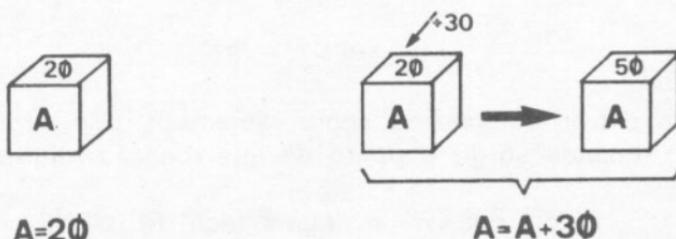
Mude agora a linha 25 para:

25 A=A+30

Pode parecer estranho para um matemático a instrução contida na linha 25. Para ser entendida, essa instrução não deve

ser encarada como uma equação e sim como uma atribuição de um novo valor à variável A. Ou seja, o computador executa a linha 25 tomando o valor antigo da variável A atribuído na linha 20 ( $A = 20$ ) e adiciona a ele mais 30 unidades. O resultado (50) é então atribuído à mesma variável: A. Isso equivale a acrescentar 30 unidades às 20 originais contidas na "caixa A". Veja a figura 4.16.

Figura 4.16 — Atribuição de valores a variáveis.



Outra forma de atribuirmos uma informação a uma variável é através do comando **INPUT**. Sua sintaxe é:

### **INPUT nome de variável**

Quando executamos um programa, o computador interrompe a execução ao "encontrar" o comando INPUT. É mostrado um ponto de interrogação e se aguarda que "alguma coisa" seja introduzida como resposta.

A resposta digitada pode ser um número ou uma seqüência de caracteres que serão atribuidos à variável associada ao comando INPUT utilizado.

Para exemplificar, digite o programa da figura 4.17.

Figura 4.17

```
10 SCREEN 0
20 INPUT "Qual o seu primeiro nome";N$
30 PRINT=PRINT"Entre dois números."
40 INPUT "Qual o primeiro";A
50 INPUT "Qual o segundo";B
60 LOCATE 0,10
70 PRINT N$; ", o produto deles é";A*B
```

O comando INPUT pode ser usado também sem qualquer texto associado. Por exemplo, as linhas 40 e 50 podem ser digitadas como:

```
40 INPUT A  
50 INPUT B
```

Nesse caso, surgem apenas os pontos de interrogação.

Além disso, podemos pedir a entrada dos dois valores num único comando INPUT. Para verificar isso, cancele a linha 50 e troque a de número 40 por:

```
40 INPUT A,B
```

Para entrar os valores agora, separe-os por vírgula. Por exemplo, quando surgir o ponto de interrogação, digite:

```
38,47 e depois tecle RETURN.
```

## OPERAÇÕES

O HOT BASIC permite que seja realizada uma série de operações com constantes ou variáveis. Essas operações podem ser divididas em quatro grupos:

- 1) Aritméticas.
- 2) Relacionais.
- 3) Lógicas.
- 4) Funcionais.

Os operadores aritméticos são, em ordem de prioridade, os relacionados na Tabela 4.2.

Para mudar a prioridade (ordem de execução), devemos utilizar parênteses.

Todas as operações aritméticas só podem ser efetuadas com constantes ou variáveis numéricas, à exceção da adição, que pode também ser utilizada para "somar" variáveis alfanuméricas (concatenação de "strings"). Por exemplo:

```
AS = "COMPUTADOR"  
BS = "HB-B000"  
AS+BS = "COMPUTADOR HB-B000"
```

As operações relacionais permitem comparação de constantes ou variáveis. Os operadores relacionais estão na Tabela 4.3.

Tabela 4.2 — Operadores Aritméticos

PRIORIDADE	OPERADOR	OPERAÇÃO	EXEMPLO	SIGNIFICADO
1	$\wedge$	Exponenciação	$8 \wedge 4$	8 elevado a 4
2	-	Mudança de Sinal	-10	-10
3	*	Multiplicação	7 * 6	7 x 6
3	/	Divisão	20 / 4	20 : 4
4	\	Divisão Inteira	10 \ 3	$10 \text{ MOD } 3 = 1$
5	MOD	Resto da Divisão	10 MOD 3	$10 \text{ MOD } 3 = 1$
6	+	Adição	10 + 5	10 + 5
6	-	Subtração	20 - 7	20 - 7

Tabela 4.3 — Operadores Relacionais

OPERADOR	OPERAÇÃO	EXEMPLO
=	Igual a	A = B
< >	diferente de	A < > B
<	menor que	A < B
>	maior que	A > B
< =	menor ou igual a	A < = B
> =	maior ou igual a	A > = B

Os operadores lógicos são usados para determinar condições múltiplas. Os principais são: OR e AND.

Para entender esses operadores imagine duas expressões que podem ser verdadeiras ou falsas. Assim:

\* (expressão 1) **OR** (expressão 2) = **VERDADEIRO** se a expressão 1 ou a expressão 2 ou as duas forem verdadeiras.

\* (expressão 1) **OR** (expressão 2) = **FALSO** se as duas expressões forem falsas.

\* (expressão 1) **AND** (expressão 2) = **VERDADEIRO** se as duas expressões forem verdadeiras.

\* (expressão 1) **AND** (expressão 2) = **FALSO** se a expressão 1 ou a expressão 2 ou as duas forem falsas.

Exemplos:

(5=5) OR (3=1) = VERDADEIRO

(5=6) OR ("ABC"="123") = FALSO

(2=2) AND ("ABC"="ABC") = VERDADEIRO

(7=7) AND (8=6) = FALSO

As operações funcionais são as que utilizam as funções do HOT BASIC tais como, LOG, SIN, CHR\$, etc. Há um número bastante grande delas e além disso, o usuário pode definir novas funções utilizando DEF FN. Para conhecê-las, consulte o Dicionário do HOT BASIC, no fim desse capítulo.

## DECISÕES

Durante a execução de um programa, muitas vezes o seu fluxo normal pode ser alterado, dependendo do resultado de algumas operações.

Para que isso seja possível, utilizamos os comandos GOTO e IF/THEN.

O comando **GOTO** ("vá para") provoca um desvio na execução do programa para a linha especificada. Por exemplo, GOTO 100 provoca um desvio para a linha de número 100 do programa.

O comando **IF/THEN** ("se/então") faz um "teste" e sua estrutura é:

**IF (a condição é verdadeira) THEN (execute um comando)**

Por exemplo,

**IF A=B THEN GOTO 100**

pode ser traduzido como: "se o conteúdo da variável A for igual ao da variável B então desvie para a linha 100 do programa".

Para ilustrar o exposto acima, analise o programa da figura 4.18.

Nesse programa, se o número introduzido for par então o resto de sua divisão por 2 é igual a zero ( $A \bmod 2 = 0$ ). Assim, ocorre desvio para a linha 60. Caso o número introduzido seja ímpar, não ocorre tal desvio e são executadas as linhas 40 e 50 do programa.

Figura 4.18

```
10 CLS
20 INPUT "Entre um número";A
30 IF A MOD 2 = 0 THEN GOTO 60
40 PRINT "O número introduzido é IMPAR"
50 GOTO 20
60 PRINT "O número introduzido é PAR"
70 GOTO 20
```

Execute esse programa comandando RUN e introduzindo alguns números. Para interromper, comande CTRL+STOP.

Modifique agora o programa cancelando as linhas 60 e 70 e alterando a de número 30 para:

```
30 IF A MOD 2 = 0 THEN PRINT "O número in
troduzido é PAR" : GOTO 20
```

O funcionamento do programa não é alterado. Note que a instrução GOTO 20 introduzida na mesma linha passa a pertencer ao teste executado por IF/THEN.

Outra mudança que pode ser feita é eliminar a linha 40 e novamente alterar a de número 30 para:

```
30 IF A MOD 2 = 0 THEN PRINT "PAR" ELSE
PRINT "IMPAR"
```

A tradução dessa linha é: "SE (IF) o resto da divisão por 2 for igual a zero ENTÃO (THEN) imprima PAR, CASO CONTRÁRIO (ELSE) imprima IMPAR".

Digite agora o programa da figura 4.19.

Figura 4.19

```
10 CLS
20 A=1
30 LOCATE A,A
40 PRINT "HB-8000"
50 A=A+1
60 IF A<21 THEN GOTO 30
70 END
```

Esse programa estabelece o que chamamos de "laço". Na linha 20 é atribuído um valor inicial à variável A (no caso, 1). Na linha 30, posiciona-se a próxima impressão de modo que tanto a coluna quanto a linha em que ela deve se iniciar correspondam ao valor de A. A linha 40 efetua a impressão. A linha 50 atribui um novo valor à variável A como sendo o seu valor atual

acrescido de uma unidade. A linha 60 verifica se a variável A é menor que 21 e se for, ocorre desvio para a linha 30, "fechando o laço" e repetindo o processo. Quando a variável A assumir o valor '21, o programa "escapa" do laço definido pelas linhas 30 a 60 e termina.

O efeito do programa acima pode também ser obtido com os comandos **FOR** e **NEXT**. Teste isso rodando o programa da figura 4.20.

Figura 4.20

```
10 CLS
20 FOR A=1 TO 20
30 LOCATE A,A
40 PRINT "HB-8000"
50 NEXT A
60 END
```

O comando **FOR** declara o nome da variável e de quanto a quanto ela deve variar. O comando **NEXT** fecha o laço. Ambos devem ser sempre utilizados conjugadamente. Não use um, sem o outro.

O incremento da variável é sempre igual a 1 a não ser que você especifique outro incremento com o comando **STEP**. Por exemplo, substitua a linha 20 por:

```
20 FOR A = 1 TO 20 STEP 2
```

Note agora que as impressões são realizadas em linhas e colunas alternadas pois a variável A assume os valores 1,3,5,...,17 e 19.

## AS TELAS GRÁFICAS

Existem mais duas telas, além das já mencionadas, no HOT BASIC. Elas permitem que sejam desenhados gráficos e figuras. Também devem ser selecionadas pelo comando **SCREEN**.

Comandando **SCREEN 2**, obtemos a tela para desenhar em Alta Resolução Gráfica. Ela contém 192 linhas e 256 colunas.

Ao executarmos **SCREEN 3**, selecionamos a tela para desenhar em Baixa Resolução Gráfica. Ela dispõe de 48 linhas e 64 colunas.

As telas gráficas são desativadas toda vez que um programa termina ou é interrompido por **CTRL+STOP**. Nesses casos, ativa-se automaticamente a **SCREEN 0** ou a **SCREEN 1**.

Quando trabalhamos com SCREEN 2 ou 3, podemos contar com vários comandos específicos para traçar pontos, linhas, círculos, podendo definir as cores de cada um.

Quanto às cores, podem ser utilizadas as 16 já apresentadas na Tabela 4.1. O comando COLOR funciona da forma já explicada com uma diferença: ele pode se referir particularmente à cor de cada figura desenhada.

Para desenhar um ponto, utilize a instrução **PSET**. Sua sintaxe é:

**PSET (x,y),c**

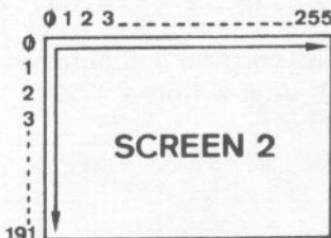
onde, x = coordenada horizontal

y = coordenada vertical

c = código da cor (0 a .15)

As coordenadas são definidas na SCREEN 2 conforme a figura 4.21.

Figura 4.21 — Coordenadas da tela definida por SCREEN 2.



Digite agora o programa da figura 4.22.

Figura 4.22

```
10 SCREEN 2
20 X=RND(1)*255 : Y=RND(1)*191 : C=INT((RND(1)*15)+1)
30 PSET (X,Y),C
40 GOTO 20
```

Deixe esse programa rodar por uns 10 minutos e veja o efeito obtido na tela. Note que, em SCREEN 2, as cores são definidas para cada 4 pontos horizontais consecutivos. Quando é dado um PSET num certo ponto com uma certa cor, os três pontos vizinhos a esse passam a ter a mesma cor. Isso não acontece na SCREEN 3. Para experimentar, mude a linha 10 para:

e verifique o resultado rodando novamente o programa. Para apagar um ponto previamente desenhado, pode-se usar a instrução PRESET. Por exemplo, PRESET(5,10) apaga o ponto desenhado por PSET(5,10)

Quando se omite a indicação da cor, a instrução PSET assume a "de frente" que foi especificada pelo comando COLOR e a instrução PRESET, por sua vez, assume a "de fundo".

Para desenhar um segmento de reta utiliza-se a instrução LINE. Sua sintaxe é:

**LINE (x1,y1)-(x2,y2),c**

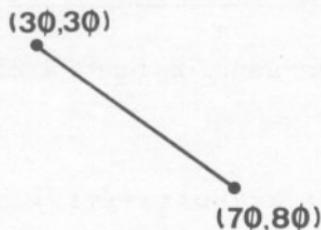
onde, (x1,y1) e (x2,y2) são as coordenadas dos dois pontos correspondentes às extremidades do segmento a ser desenhado e c é a cor desse segmento.

Por exemplo,

**LINE (30,30)-(70,80),7**

desenha um segmento na cor azul anil entre os pontos de coordenadas (30,30) e (70,80). Veja a figura 4.23.

Figura 4.23 — Linha traçada pelo comando LINE.

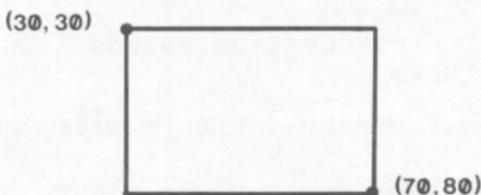


Para traçar um retângulo, colocamos a letra B ("Box") no fim da instrução. Nesse caso, as coordenadas indicam os vértices superior esquerdo e inferior direito do retângulo. Por exemplo,

**LINE (30,30)-(70,80),8,B**

desenha um retângulo vermelho conforme o indicado na figura 4.24.

Figura 4.24 — Retângulo desenhado pelo comando LINE acompanhado da letra B.



Para colorir a parte interna do retângulo, usam-se, no fim da instrução, as letras **BF** ("Box Fill"). Por exemplo,

**LINE (30,30)-(70,80),8,BF**

desenha um retângulo vermelho totalmente pintado. O programa da figura 4.25 desenha linhas em forma de grelha com as cores variando.

Figura 4.25

```
10 COLOR 15,1,7 = SCREEN 2
20 C=1
30 FOR X=0 TO 255 STEP 4
40 C=C+1
50 IF C>=15 THEN C=2
60 LINE (X,1)-(X,190),C
70 NEXT X
80 GOTO 30
```

Para traçar circunferências, elipses ou arcos, utilize a instrução **CIRCLE**. A sua sintaxe é:

**CIRCLE (x,y),r,c,ai,af,a**

onde, (x,y) = coordenadas do centro

r = raio

c = cor

ai = ângulo inicial do arco (radianos)

af = ângulo final do arco (radianos)

$a =$  relação entre o eixo vertical e o eixo horizontal.  
Lembre-se que uma circunferência completa tem  $6.28$  radianos.

Por exemplo, querendo traçar uma circunferência com o seu centro localizado nas coordenadas (128,90) e de raio igual a 50, devemos comandar:

**CIRCLE (128,90),50**

Se você quiser especificar a cor, acrescente o código relativo a ela.

Digite o programa exemplo da figura 4.26.

Figura 4.26

```
10 SCREEN 2 : COLOR 15,1,1
20 FOR X=10 TO 255 STEP 5
30 CIRCLE (X,80),40,8
40 NEXT X
50 GOTO 50
```

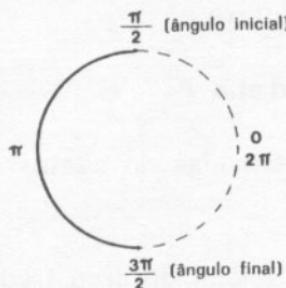
Com a instrução CIRCLE, podemos traçar arcos especificando os ângulos iniciais e finais medidos em radianos. Para exemplificar, rode o programa da figura 4.27.

Figura 4.27

```
10 SCREEN 2 : COLOR 15,1,1
20 PI=4*ATN(1)
30 CIRCLE (128,90),70,7,PI/2,3*PI/2
40 GOTO 40
```

Na linha 20, definimos a variável PI (3.1416) utilizando a função arco-tangente. A linha 30 traça um arco de circunferência na cor azul (7) conforme a figura 4.28.

Figura 4.28 — Arco de circunferência traçado pela instrução CIRCLE.



Colocando o sinal de "menos" antes dos valores dos ângulos inicial e final, serão traçados setores circulares. Experimente substituir a linha 30 do programa da figura 4.27 por:

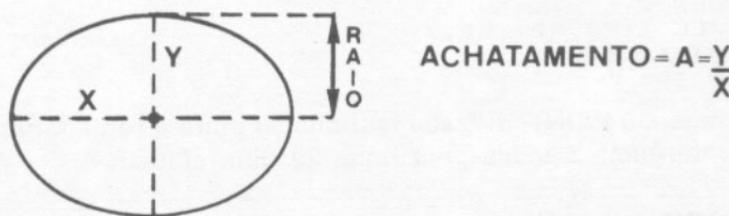
```
30 CIRCLE (128,90),70,7,-PI/2,-3*PI/2
```

Quando os valores dos ângulos inicial e final são omitidos, o computador assume 0 e 6.28 ( $2*\pi$ ), respectivamente. Dessa forma é obtida a circunferência completa.

Nos exemplos citados até agora, você deve ter notado que as circunferências traçadas não são exatamente "redondas"! Isso ocorre pois a "largura" é maior que a "altura" nos pontos traçados na tela. Para obter uma circunferência "redonda" acrescentamos mais um parâmetro para a instrução CIRCLE: o "achatamento".

O "achatamento" pode ser entendido como a razão entre os eixos vertical e horizontal. Veja a figura 4.29.

Figura 4.29 — Achatamento



Fazendo o "achatamento" ser igual a 1.25, obtém-se a "tão desejada circunferência redonda"! Nesse caso, o raio é referente à medida do semi-eixo vertical (figura 4.29). Para verificar isso, digite o programa da figura 4.30.

Figura 4.30

```
10 SCREEN 2 : COLOR 15,1,1
20 CIRCLE (128,80),60,7,,,1.25
30 GOTO 30
```

Note na linha 20 que, omitindo alguns parâmetros, as vírgulas relativas a eles não devem ser omitidas.

Altere agora o "achatamento" trocando a linha 20 por:

```
20 CIRCLE (128,80),60,7,,,0.8  
ou  
20 CIRCLE (128,80),60,7,,,3
```

Você deverá obter elipses conforme as da figura 4.31.

Figura 4.31 — Elipses



ACHAT < 1.25



ACHAT > 1.25

Digite agora o programa da figura 4.32.

Figura 4.32

```
10 SCREEN 2  
20 CIRCLE (128,80),50,7  
30 PAINT (128,80),7  
40 GOTO 40
```

O comando **PAINT** utilizado na linha 30 pinta a região interior da circunferência traçada pela linha 20. Sua sintaxe é:

**PAINT (x,y),c**

onde, (x,y) = coordenadas de um ponto qualquer pertencente à região a ser pintada.

c = cor da tinta utilizada na pintura.

O microcomputador considera a região a ser pintada como sendo aquela delimitada por uma linha fechada (fronteira) à qual pertence o ponto definido pelas coordenadas do comando **PAINT**.

Quanto à cor utilizada por **PAINT**, deve ser a mesma da linha que define a fronteira quando estivermos utilizando **SCREEN 2**. No modo **SCREEN 3** esse cuidado não é necessário.

Para desenhar quaisquer figuras, o HB-8000 possui uma função extremamente poderosa chamada **DRAW** ("desenhe"). Sua

sintaxe é:

### DRAW string

onde, string é uma cadeia de caracteres ou uma variável alfanumérica contendo os códigos da Tabela 4.4.

Tabela 4.4 — Algumas instruções macro-gráficas do comando DRAW.

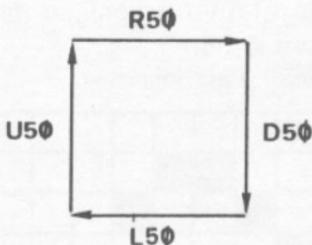
Un	traça uma linha para cima da posição fixada para o cursor.
Dn	traça uma linha para baixo.
Ln	traça uma linha para esquerda.
Rn	traça uma linha para direita.
En	traça uma linha diagonalmente para cima e direita.
Fn	traça uma linha diagonalmente para baixo e direita.
Gn	traça uma linha diagonalmente para baixo e esquerda.
Hn	traça uma linha diagonalmente para cima e esquerda.
Cn	especifica a cor (n = 0 a 15).
Sn	especifica a escala.
B	permite movimento sem traçar qualquer linha.
N	permite traçar linha sem movimento do cursor.

Para exemplificar, o comando

**DRAW "R50D50L50U50"**

deve traçar um quadrado conforme o da figura 4.33.

Figura 4.33 — Quadrado traçado pelo comando DRAW.



Digite agora o programa da figura 4.34 e veja o seu efeito.

Figura 4.34

```
10 SCREEN 2 : COLOR 15,1,1
20 PSET (128,80),0
30 DRAW "C7NU70ND70NL70NR70NE70NF70NG70N
H70"
40 DRAW "BH70R140D140L140U140"
50 GOTO 50
```

## TEXTO IMPRESSO EM TELA GRÁFICA

No modo gráfico o comando PRINT não funciona para imprimir os caracteres na tela.

Para se visualizar os caracteres em SCREEN 2 ou 3, devemos usar as instruções:

**OPEN "GRP:" FOR OUTPUT AS #1**  
e  
**PRINT #1, "texto a ser impresso"**

Para posicionar o início do texto na tela devemos utilizar os comandos .PSET ou PRESET (e não LOCATE).

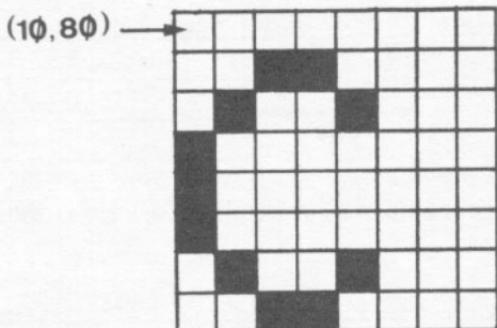
Veja o programa da figura 4.35 para ver como posicionar, imprimir e depois apagar um texto em SCREEN 2.

Figura 4.35

```
10 SCREEN 2 : COLOR 7,1,1
20 OPEN "GRP:" FOR OUTPUT AS #1
30 PRESET(10,80)
40 PRINT #1,"COMPUTADOR HOTBIT"
50 FOR I= 1 TO 2000 : NEXT I
60 LINE (10,80)-(89,87),1,BF
70 GOTO 70'
```

A linha 30 indica o ponto referente ao vértice superior esquerdo do campo 8x8 relativo à posição da primeira letra a ser impressa. Veja a figura 4.36.

Figura 4.36 — Primeira letra a ser impressa.

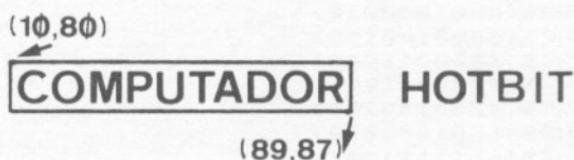


A linha 40 efetua a impressão e a de número 50 determina um intervalo de tempo com um "laço vazio".

A linha 60 desenha um retângulo sobre a palavra COMPUTADOR.

TADOR e o pinta na cor de fundo (no caso, preto = 1). Veja a figura 4.37.

Figura 4.37 — Apagando um texto impresso em SCREEN 2.



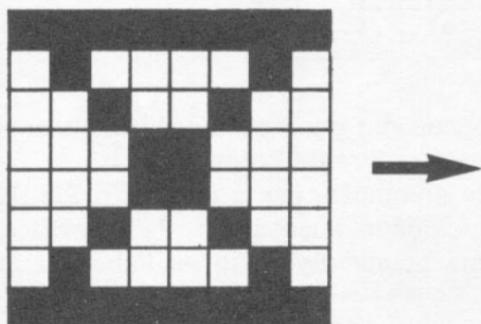
## MOVIMENTAÇÃO DE UMA FIGURA - SPRITE

O HOT BASIC permite desenhar na tela figuras que podem movimentar-se sem interferir nos textos ou figuras já desenhadas. Essas figuras, passíveis de animação, são chamadas de "SPRITES" e podem ser usadas em SCREEN 1, 2 ou 3.

Para utilizar os "SPRITES", deve-se inicialmente determinar a forma das figuras a elas associadas. Apesar de um "SPRITE" ser definido nos formatos 8x8 e 16x16, vejamos como fazê-lo no formato 8x8.

Inicialmente, faça uma figura num quadriculado 8x8 e associe a cada ponto "cheio" o valor 1 e a cada ponto "vazio" o valor 0. Veja um exemplo na figura 4.38.

Figura 4.38 — Definindo um "SPRITE".



A seguir, utiliza-se a variável **SPRITES\$ (n)**, onde "n" deve ser o número do SPRITE e estar compreendido entre 0 e 255.

Defina-o conforme o programa da figura 4.39.

Figura 4.39

```
10 SCREEN 2: COLOR 15,1,1
20 A$(1)=CHR$(&B11111111)
30 A$(2)=CHR$(&B01000010)
40 A$(3)=CHR$(&B00100100)
50 A$(4)=CHR$(&B00011000)
60 A$(5)=CHR$(&B00011000)
70 A$(6)=CHR$(&B00100100)
80 A$(7)=CHR$(&B01000010)
90 A$(8)=CHR$(&B11111111)
100 FOR I=1 TO 8: B$=B$+A$(I): NEXT I
110 SPRITE$(1)=B$
```

Para que apareça o SPRITE definido na tela, devemos usar a instrução PUT SPRITE, cuja sintaxe é:

**PUT SPRITE p,(x,y),c,n**

onde, p = número do plano ou camada (0 a 31).

(x,y) = coordenadas do vértice superior esquerdo do SPRITE.

c = cor do SPRITE.

n = número do SPRITE.

No programa anterior, acrescente as linhas da figura 4.40.

Figura 4.40

```
120 PUT SPRITE 0,(110,60),8,1
130 PUT SPRITE 1,(120,70),7,1
200 GOTO 200
```

O mesmo SPRITE é colocado em duas camadas, posições e cores diferentes na tela com o programa anterior.

Para produzir o efeito de animação, basta colocar o SPRITE numa MESMA camada em posições sucessivas. Para verificar este efeito, altere o programa acima digitando as linhas da figura 4.41.

Note que, quando é colocado o SPRITE numa mesma camada mas numa posição diferente, o que foi colocado na posição anterior é apagado automaticamente..

Figura 4.41

```
120 C=2
130 FOR X=10 TO 250
140 PUT SPRITE 0,(X,60),C,1
150 NEXT X
160 C=C+1:IF C>15 THEN C=2
170 GOTO 130
```

Resumindo, as regras que estabelecem o uso dos SPRITES são:

- 1) Os SPRITES podem ser colocados em SCREEN 1, 2 ou 3.
- 2) Não é possível colocar-se mais de um SPRITE simultaneamente na mesma camada.
- 3) São utilizadas 32 camadas, portanto, apenas 32 SPRITES podem aparecer simultaneamente na tela.
- 4) É possível inserir simultaneamente numa linha horizontal da tela um máximo de 4 SPRITES, em camadas diferentes.
- 5) Quando se sobreponem dois SPRITES de diferentes camadas, o que estiver na camada de menor número se sobrepõe ao outro.

## SONS

O HB-8000 possui um Gerador Programável de Sons (PSG) que permite facilmente, através do comando **PLAY**, produzir sons e músicas.

Os sons no comando **PLAY** podem ser produzidos em três canais independentes e são especificados por subcomandos representados por "strings" dentro de " " ou por variáveis "string".

A sintaxe do comando **PLAY** é:

**PLAY string 1, string 2, string 3**

onde, string 1 = subcomandos do canal 1.

string 2 = subcomandos do canal 2.

string 3 = subcomandos do canal 3.

Os subcomandos utilizados pelo comando **PLAY** estão relacionados na Tabela 4.5.

As notas musicais são indicadas de acordo com a notação cifrada. As notas e os semi-tonos são especificados na figura 4.42.

Figura 4.42 — Escala musical.

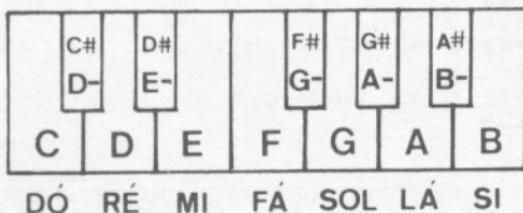
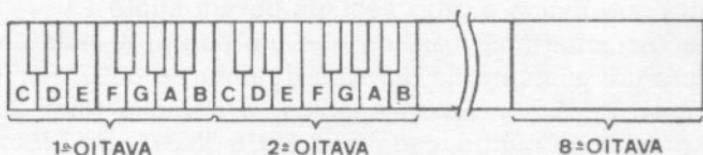


Tabela 4.5 — Subcomandos da função PLAY

SUBCOMANDO	VALORES PERMITIDOS	FINALIDADE
Tn	n = 32 a 255	Determinar o tempo (andamento) da música.
On	n = 1 a 8	Determinar uma das 8 oitavas do HOT BASIC.
Ln	n = 1 a 64	Determinar a duração da nota.
Nn	n = 0 a 96	Especificar uma nota musical através de sua numeração absoluta.
A,B,C,D,E,F,G		Indicar a nota musical dentro de uma oitava pré-determinada.
# ou +		Sustenido
-		Bemol
Rn	n = 1 a 64	Determinar uma pausa.
. (ponto)		Aumentar a duração de uma nota ou pausa em 50%.
Vn	n = 0 a 15	Determinar o volume.
Mn	n = 0 a 65535	Determinar o período da variação do volume durante a execução de uma nota.
Sn	n = 0 a 15	Determinar a forma de variação do volume durante a execução de uma nota.

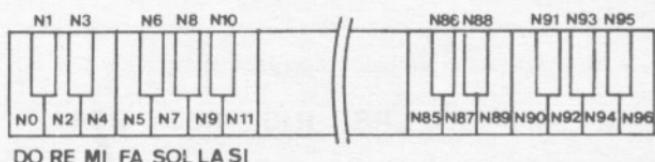
A oitava é selecionada por **On**, onde n pode variar de 1 a 8. Na ausência dessa indicação, o computador assume n=4 (quarta oitava). A figura 4.43 mostra a correspondência entre as oitavas e o teclado de um piano.

Figura 4.43 — Correspondência entre as oitavas e o teclado de um piano.



A notação **Nn** permite usar uma numeração absoluta referente ao teclado musical completo. O número n deve estar compreendido entre 0 e 96. A correspondência entre os números das notas e as teclas de um piano é mostrado na figura 4.44.

Figura 4.44 — Correspondência entre os números e as teclas de um piano.



A notação **Ln** especifica a duração de uma nota, onde, "n" é um número inteiro compreendido entre 1 e 64 e o seu valor tem o significado apresentado na Tabela 4.6.

Tabela 4.6 — Duração de uma nota.

L1 ○	n = 1 , indica uma nota semibreve (nota de maior duração)
L2 ♩	n = 2 , indica uma nota mínima (2 mínimas = 1 semibreve)
L4 ♪	n = 4 , indica uma nota semimínima (4 semimínimas = 1 semibreve)
L8 ♫	n = 8 , indica uma nota colchela (8 colchelas = 1 semibreve)
L16 ♩	n = 16 , indica uma nota semicolchela (16 semicolchelas = 1 semibreve)
L32 ♫	n = 32 , indica uma nota fusa (32 fusas = 1 semibreve)
L64 ♩	n = 64 , indica uma nota semifusa (64 semifusas = 1 semibreve)

Assim como nas partituras musicais, para indicar uma duração de uma vez e meia a duração original da nota, basta escrever a letra que indica a nota seguida de um ponto (.).

Se acrescentarmos mais do que um ponto, a cada um corresponderá um aumento na duração da nota de 50%. Por exemplo, no caso de 3 pontos sucessivos, a duração final da nota é igual à original multiplicada por 3.375 ( $3.375 = 1.5 * 1.5 * 1.5$ ).

Para indicarmos uma pausa musical utilizamos **Rn**, onde "n" é um número inteiro compreendido entre 1 e 64. O critério para a duração de uma pausa é o mesmo que o da duração de uma nota. Isto é, R1 indica uma pausa equivalente à duração de uma semibreve (L1), R2 indica uma pausa que equivale à duração de uma breve (L2), e assim por diante até R64.

A figura 4.45 mostra as pausas e seus símbolos correspondentes.

Figura 4.45 — Pausas e seus símbolos correspondentes.

R1	R2	R4	R8	R16	R32	R64

Como no caso das notas, o ponto aumenta a duração da pausa em 50%.

O tempo da música é especificado por **Tn**, onde "n" pode variar de 32 a 255. Quanto maior for "n", mais rapidamente será executada a música. Na ausência de indicação de tempo, o valor assumido é 120.

O volume do som é dado por **Vn**, onde, "n" pode valer algo entre 0 e 15, sendo o seu valor inicial igual a 8.

Depois do exposto até aqui, temos condições de transcrever notas de um pentagrama para a linguagem musical do HOT BASIC. Para exemplificar, veja a figura 4.46.

Figura 4.46 — Transcrição das notas musicais de um pentagrama.

PLAY" L404G. " equivale a PLAY" L4N43. "

PLAY" L804D+ " equivale a PLAY" L8N39"

É possível suprimir a letra L e escrever a duração da nota (n) após o seu nome. Assim:

PLAY "L405A" equivale a PLAY "05A4"

A duração (L) e a oitava (O), se forem suprimidas, assumem o último valor especificado.

Para finalizar e exemplificar o comando PLAY, execute o programa da figura 4.47.

Figura 4.47

```
100 #####H#####H#####H#####H#####H#####H#####H#####
110 '# Turkey in the straw #
130 #####
140 PLAY"T250S0M5000","T250S0M5000","T25
0S0M5000"
150 '
160 PLAY"o4E8D8","R4","R4"
170 PLAY"C4L8CDCo3GEF","o4L4CGEG","o5R4L
4CR4C"
180 PLAY"GAGEG4o4C8D8","CGEG","R4CR4C"
190 PLAY"E4E4L8EDCD","CGEG","R4CR4C"
200 '
210 PLAY"E4D4R4E8D8","o3Go4Do3Bo4D","R4o
4GR4G"
220 PLAY"C4L8CDCo3GEF","CGEG","R4o5CR4C"
230 PLAY"GAGEG4o4C8D8","CGEG","R4CR4C"
240 PLAY"E8G4L8AGECD","CGEG","R4CR4C"
250 '
260 PLAY"L4EDCR4","Co3Bo4CR4","o4GGGR4"
270 PLAY"E8G4E8G4G4","CGEG","R4o5CR4C"
280 PLAY"E8G4E8G4G4","CGEG","R4CR4C"
290 PLAY"F8A4F8A4A4","o3Go4Co3Ao4C","R4o
4FR4F"
300 PLAY"F8A4F8A4o3A8B8","o3Go4Co3Ao4C",
"R4FR4F"
310 '
320 PLAY"o4L4CCo3GG","CGEG","R4o5CR4C"
330 PLAY"EEDL8CD","DHR4o3BR4","o4A#R4GR4
"
340 PLAY"EG4L8AGECD","o4CR4CR4","GR4GR4"
350 PLAY"L4EDCR1","Co3Bo4CR4","GGGR4"
```

Experimente agora, fazer a alteração da figura 4.48.

Figura 4.48

```
140 PLAY"T250S13M700","T250S13M700","T25
0S13M700"
```

## DICIONÁRIO DO HOT BASIC

<b>ABS</b>	Função que fornece o valor absoluto de um número. Formato: ABS (argumento)
<b>AND</b>	Operação lógica "e".
<b>ASC</b>	Função que fornece o código ASCII do primeiro caracter de uma "string". Formato: ASC (string não nula)
<b>ATN</b>	Função que fornece o valor do arco, em radianos cuja tangente trigonométrica é igual ao argumento. Formato: ATN (argumento)
<b>AUTO</b>	Comando que gera, automaticamente, números de linhas de um programa a partir de um valor inicial especificado, com incremento também especificado. Formato: AUTO AUTO número da linha inicial, incremento
<b>BASE</b>	Comando que permite o acesso ao endereço inicial das tabelas utilizadas pelo processador de vídeo. Formato: BASE (número inteiro entre 0 e 19) BASE (número) = número
<b>BEEP</b>	Comando que provoca a emissão de um sinal sonoro. Formato: BEEP
<b>BINS</b>	Função que transforma um dado numérico em uma expressão binária na forma de uma string. Formato: BINS (argumento)
<b>BLOAD</b>	Comando que carrega um arquivo em binário e o executa, caso esta opção esteja ativada com a letra R. Formato: BLOAD "nome do dispositivo: nome do arquivo", R, deslocamento
<b>BSAVE</b>	Comando que grava programas em binário uma área específica da memória. Formato: BSAVE "nome do dispositivo: nome do arquivo", endereço inicial, endereço final, endereço do início da execução
<b>CALL</b>	Executa uma rotina em linguagem de máquina. Formato: CALL nome do comando, argumentos
<b>CDBL</b>	Função que converte um dado numérico no formato "simples precisão" em um dado no formato de "dupla precisão". Formato: CDBL (argumento)

<b>CHR\$</b>	Função que fornece o caracter correspondente ao código especificado. Formato: CHR\$ (argumento inteiro entre 0 e 255)
<b>CINT</b>	Função que converte dados numéricos em números inteiros. Formato: CINT (argumento entre -32768 e 32767)
<b>CIRCLE</b>	Traça, no modo gráfico, um círculo, uma elipse, uma parte de um arco circular ou um setor. Formato: CIRCLE (x,y), raio, cor, ângulo inicial, ângulo final, proporção entre eixo Y e X
<b>CLEAR</b>	Comando que inicializa todas as variáveis e estabelece o tamanho da área de caracteres e o último endereço de memória utilizada pelo BASIC. Também fecha todos os arquivos abertos. Formato: CLEAR tamanho da área de caracteres, último endereço da RAM utilizável pelo BASIC.
<b>CLOAD</b>	Carrega um programa em HOT BASIC de uma fita cassete para a memória do micro. Formato: CLOAD "nome do arquivo"
<b>CLOAD?</b>	Compara um programa BASIC da fita cassete com o da memória. Formato: CLOAD? "nome do arquivo"
<b>CLOSE</b>	Fecha um arquivo que foi aberto por um comando OPEN. Formato: CLOSE # número de arquivo aberto por OPEN
<b>CLS</b>	Comando que apaga tudo o que estiver visualizado na tela. Formato: CLS
<b>COLOR</b>	Especifica as cores do primeiro plano, do fundo e das bordas. Formato: COLOR cor do primeiro plano, cor do fundo, cor da borda (onde, cor = número inteiro entre 0 e 15)
<b>CONT</b>	Continua a execução de um programa interrompido. Formato: CONT
<b>COS</b>	Função que fornece o valor do cosseno de um ângulo em radianos. Formato: COS (ângulo)

<b>CSAVE</b>	Armazena em fita cassete, um programa em HOT BASIC. Formato: CSAVE "nome do arquivo", velocidade de transmissão em bauds
<b>CSNG</b>	Função que converte um dado numérico para precisão simples. Formato: CSNG (argumento)
<b>CSRLIN</b>	Fornece a linha em que se encontra o cursor. Formato: CSRLIN
<b>DATA</b>	Fornece os dados que serão lidos pelo comando READ. Formato: DATA constante, constante, ...
<b>DEF FN</b>	Define e atribui um nome a uma função escrita pelo usuário. Formato: DEF FN nome da função (lista de parâmetros) = (expressão da função definida)
<b>DEFDBL</b>	Declara quais as letras iniciais dos nomes das variáveis que armazenarão números de precisão dupla: Formato: DEFDBL caracter(es)
<b>DEFINT</b>	Declara quais as letras iniciais dos nomes das variáveis que armazenarão números inteiros. Formato: DEFINT caracter(es)
<b>DEFSNG</b>	Declara quais as letras iniciais dos nomes das variáveis que armazenarão números de precisão simples. Formato: DEFSNG caracter(es)
<b>DEFSTR</b>	Declara quais as letras iniciais dos nomes das variáveis que armazenarão strings. Formato: DEFSTR caracter(es)
<b>DEFUSR</b>	Define um endereço para iniciar a execução de uma sub-rotina em linguagem de máquina. Formato: DEFUSR dígito = endereço para início da execução
<b>DELETE</b>	Apaga as linhas de um programa. Formato: DELETE número da linha A — número da linha B
<b>DIM</b>	Definição de uma ou mais variáveis do tipo matriz, especificando o tipo de matriz e sua dimensão. Formato: DIM nome da variável (valor máximo de um sub-índice,...)
<b>DRAW</b>	Desenha figuras na tela gráfica (SCREEN 2 ou 3), de

	acordo com o sub-comando especificado.
<b>END</b>	Formato: DRAW "subcomandos" Termina a execução de um programa, fecha todos os arquivos e retorna ao estado de espera de um comando direto.
<b>EOF</b>	Formato: END Utilizado para testar o fim de um arquivo.
<b>ERASE</b>	Formato: EOF (número do arquivo aberto por OPEN) Apaga, da memória, as variáveis do tipo matriz especificadas na instrução.
<b>ERL</b>	Formato: ERASE nome da variável Função que fornece o número da linha do programa em que ocorreu um erro.
<b>ERR</b>	Formato: ERL Função que fornece o número de um erro ocorrido.
<b>ERROR</b>	Formato: ERR Simula a ocorrência de um erro e permite que o usuário defina um código de erro correspondente.
<b>EXP</b>	Formato: ERROR número do erro entre 0 e 255 Função que fornece o valor de e elevado ao argumento fornecido.
<b>FIX</b>	Formato: EXP (argumento menor que 145.0628) Formato: FIX (argumento) Função que fornece a parte inteira de um dado numérico.
<b>FOR-NEXT</b>	Formato: FIX (argumento) Repete a execução de um bloco de instruções entre um comando FOR e o seu correspondente NEXT. Formato: FOR variável = valor inicial TO valor final NEXT variável
<b>FRE</b>	Formato: FOR variável = valor inicial TO valor final NEXT variável Fornece o número de bytes livres na RAM de uma área de memória não utilizada pelo HOT BASIC. Formato: FRE (0) para bytes livres para o BASIC FRE (" ") para bytes livres para strings
<b>GOSUB</b>	Transfere a execução do programa para a sub-rotina especificada e indica o ponto de retorno ao programa principal. Formato: GOSUB número da linha RETURN
<b>GOTO</b>	Desvia incondicionalmente a seqüência normal do programa para uma linha especificada. Formato: GOTO número da linha

<b>HEX\$</b>	Função que transforma um número em uma expressão hexadecimal. Formato: HEX\$ (argumento inteiro entre -32768 e e 65535)
<b>IF-THEN-ELSE</b>	Permite a bifurcação na execução do programa em função do cumprimento de uma condição. Formato: IF condição THEN comando se condição verdadeira ELSE comando se condição falsa
<b>INKEY\$</b>	Função que permite obter o caractere da tecla pressionada. Formato: INKEY\$
<b>INP</b>	Lê os dados da via de acesso especificada. Formato: INP (número da via de acesso entre 0 e 255)
<b>INPUT</b>	Introduz o valor de uma variável através do teclado. Formato: INPUT variável INPUT "mensagem", variável
<b>INPUT #</b>	Lê dados de um arquivo aberto, associando-os às variáveis de um programa. Formato: INPUT # número do arquivo, lista de variáveis
<b>INPUT\$</b>	Lê um número especificado de caracteres introduzidos pelo teclado ou através de um arquivo. Formato: INPUT\$ (número de caracteres) INPUT\$ (número de caracteres, # número do arquivo)
<b>INSTR</b>	Função que localiza a posição de uma string dentro de uma outra string. Formato: INSTR (string de pesquisa, string procurada)
<b>INT</b>	Função que fornece o valor do maior inteiro possível, menor do que o argumento. Formato: INT (argumento)
<b>INTERVAL IN/OFF/ STOP</b>	Cada uma destas instruções respectivamente, habilita, desabilita e adia uma interrupção feita pelo temporizador. Formato: INTERVAL ON INTERVAL OFF INTERVAL STOP

<b>KEY</b>	Associa uma string (cadeia de caracteres) a uma tecla de função (F1 a F10). Formato: KEY número da tecla de função, string
<b>KEY LIST</b>	Visualiza o conteúdo das teclas de função. Formato: KEY LIST
<b>KEY ON/ OFF</b>	Ativa ou desativa a visualização, na parte inferior da tela, dos caracteres associados às teclas de função. Formato: KEY ON KEY OFF
<b>KEY (n) ON/OFF/ STOP</b>	Cada uma dessas três instruções respectivamente, habilita, desabilita e adia uma interrupção através das teclas de função. Formato: KEY (n) ON KEY (n) OFF                                 (n = 1 a 10) KEY (n) STOP
<b>LEFT\$</b>	Função que fornece na forma de string os n primeiros caracteres de uma cadeia. Formato: LEFT\$ (string, quantidade de caracteres)
<b>LEN</b>	Função que fornece a quantidade de caracteres de uma string. Formato: LEN (string)
<b>LET</b>	Realiza a associação de um dado a uma variável Formato: LET variável = valor ou expressão
<b>LINE</b>	Traça, no modo gráfico, uma linha ou um quadrado, dependendo da instrução. Formato: LINE (x1,y1)-(x2,y2),cor LINE (x1,y1)-(x2,y2),cor,B LINE (x1,y1)-(x2,y2),cor,BF
<b>LINE</b>	Associa um dado introduzido pelo teclado, a uma variável string.
<b>INPUT</b>	Formato: LINE INPUT "mensagem"; variável Lê uma seqüência de até 254 caracteres de um arquivo e a atribui a uma variável string
<b>LINE INPUT #</b>	Formato: LINE INPUT # número do arquivo, nome da variável
<b>LIST/ LLIST</b>	Lista o programa ou parte dele na tela ou na impressora. Formato: LIST número da linha inicial-número da linha final

<b>LOAD</b>	Carrega um arquivo de um dispositivo especificado para a memória do microcomputador. Formato: LOAD "nome do dispositivo: nome do arquivo"
<b>LOCATE</b>	Move o cursor para posição especificada pelas coordenadas x e y. Formato: LOCATE x,y
<b>LOG</b>	Função que determina o logaritmo natural (ln) de um número. Formato: LOG (X) (X maior que zero)
<b>LPOS</b>	Função que fornece a posição do cabeçote da impressora. Formato: LPOS (0)
<b>LPRINT</b>	Escreve na impressora o valor de uma expressão. Formato: LPRINT "expressão"
<b>LPRINT USING</b>	Escreve na impressora, com o formato especificado, o valor de uma expressão. Formato: LPRINT USING "símbolo de formato" expressão
<b>MAXFILES</b>	Determina o número de arquivos que podem ser abertos ao mesmo tempo em um programa. Formato: MAXFILES = número de arquivos (0 a 15)
<b>MERGE</b>	Carrega um programa armazenado em código ASCII e o une com o programa que está na memória do computador. Formato: MERGE "nome do dispositivo nome do arquivo"
<b>MID\$</b>	Função que seleciona parte de strings (cadeias alfanuméricas). Formato: MID\$ (A\$,m,n)
<b>MID\$=</b>	Substitui uma parte de uma string por elementos de outra string. Formato: MID\$(A\$,m,n)=B\$
<b>MOTOR</b>	Liga ou desliga o motor do gravador. Formato: MOTOR ON MOTOR OFF
<b>NEXT</b>	Veja a instrução FOR.
<b>NEW</b>	Apaga o programa da memória. Formato: NEW
<b>NOT</b>	Operação lógica "não".
<b>OCTS</b>	Função que transforma um dado numérico em uma

	string na forma octal. Formato: OCT\$ (argumento entre -32768 e 65535)
<b>ON ERROR</b>	Desvia o programa para uma determinada linha quando ocorre um erro na execução ou na entrada de dados.
	Formato: ON ERROR GOTO número da linha
<b>ON-GOSUB</b>	Desvia o programa para uma sub-rotina condicionada pelo valor de uma variável.
	Formato: ON variável GOSUB número de linha, número de linha, ...
<b>ON-GOTO</b>	Desvio condicionado por uma variável.
	Formato: ON variável GOTO número de linha, número de linha, ...
<b>ON INTERVAL GOSUB</b>	Declara para qual sub-rotina deve ser desviada periodicamente a execução do programa em um intervalo de tempo especificado.
	Formato: ON INTERVAL = intervalo GOSUB número de linha
<b>ON KEY-GOSUB</b>	Declara para qual sub-rotina deve ser desviada a execução do programa no caso de pressionar-se alguma tecla de função.
	Formato: ON KEY GOSUB número de linha para F1, número de linha para F2, número de linha para F3, ...
<b>ON SPRITE-GOSUB</b>	Declara para qual sub-rotina deve ser desviada a execução do programa no caso de dois "sprites" se sobreponem.
	Formato: ON SPRITE GOSUB número de linha
<b>ON STOP GOSUB</b>	Declara para qual sub-rotina deve ser desviada a execução do programa caso se pressione CTRL+STOP.
	Formato: ON STOP GOSUB número de linha
<b>ON STRIG GOSUB</b>	Declara para qual sub-rotina deve ser desviada a execução do programa no caso da barra de espaços ou os botões dos joysticks serem pressionados.
	Formato: ON STRIG GOSUB número de linha para barra de espaços, número de linha para botão, número de linha para botão, ...
<b>OPEN</b>	Abre um arquivo e especifica o modo leitura ou escrita.

	Formato: OPEN "nome do dispositivo nome do arquivo" FOR modo AS # número do arquivo onde, modo = OUTPUT ou INPUT núm. do arquivo = 1 até o especificado por MAXFILES
<b>OR</b>	Operação lógica "ou".
<b>OUT</b>	Coloca um byte na porta especificada. Formato: OUT número da porta entre 0 e 255, expressão inteira entre 0 e 255
<b>PAD</b>	Função que verifica o estado do "touch pad". Formato: PAD (n) (n=0 a 7)
<b>PAINT</b>	Preenche uma certa área com uma cor especificada pelo usuário. Formato: PAINT (x,y),cor
<b>PDL</b>	Função que apresenta o valor determinado pelo "paddle". Formato: PDL (n) (n=0 a 12)
<b>PEEK</b>	Função que apresenta o valor armazenado em um endereço da memória. Formato: PEEK (endereço de -32768 a 65535)
<b>PLAY</b>	Toca seqüências de notas e/ou acordes musicais compostos de uma a três notas simultaneamente, com tempo, oitava, duração, tom e volume programáveis. Formato: PLAY subcomandos
<b>POINT</b>	Função que permite obter o código da cor de um ponto especificado na tela. Formato: POINT (coluna,linha)
<b>POKE</b>	Escreve um dado numérico em um certo endereço da memória. Formato: POKE endereço entre -32768 e 65535, expressão inteira entre 0 e 255.
<b>POS</b>	Função que indica a abcissa x que o cursor ocupa Formato: POS (0)
<b>PRESET</b>	Acende ou apaga um ponto nas telas gráficas. Formato: PRESET (x,y),cor
<b>PRINT</b>	Apresenta dados na tela. Formato: PRINT expressão
<b>PRINT USING</b>	Apresenta dados com um formato específico na tela. Formato: PRINT USING formato, expressão

<b>PRINT #</b>	Escreve em arquivo aberto por OPEN. Formato: PRINT # número de arquivo, expressão
<b>PRINT #</b> <b>USING</b>	Escreve dados com formato desejado em arquivo aberto pelo comando OPEN. Formato: PRINT # número de arquivo, USING símbolo de formato, expressão
<b>PSET</b>	Desenha um ponto nas telas gráficas. Formato: PSET (x,y),cor
<b>PUT</b> <b>SPRITE</b>	Torna visível um "sprite" especificado em uma posição, plano e cor especificados. Formato: PUT SPRITE número do plano de 0 a 31, (x,y), cor, número do sprite
<b>READ</b>	Lê os dados que foram relacionados em uma ou várias instruções DATA. Formato: READ variável, variável, ...
<b>REM</b>	Introdução de comentários ou observações na listagem do programa. Formato: REM comentário
<b>RENUM</b>	Renumera as linhas do programa. Formato: RENUM novo número da primeira linha, número da linha a partir da qual será renumerado, incremento
<b>RESTORE</b>	Indica a partir de qual linha a instrução READ deve ler os dados contidos na instrução DATA. Formato: RESTORE número de linha
<b>RESUME</b>	Indica a linha de retorno da rotina principal, após a execução da rotina de erro. Formato: RESUME RESUME NEXT RESUME número de linha
<b>RETURN</b>	Determina o ponto de retorno de uma sub-rotina. Veja o comando GOSUB. Formato: RETURN RETURN número de linha do programa
<b>RIGHT\$</b>	Função que separa um pedaço de uma string, começando pela direita. Formato: RIGHTS\$ (string, número de caracteres que serão destacados)
<b>RND</b>	Função que gera um valor pseudo-aleatório entre 0

e 1.

Formato: RDN (X) (onde, X = -32768 a 32767)  
RND (-TIME)

**RUN** Inicia a execução de um programa a partir de uma linha especificada.

Formato: RUN  
RUN número de linha

**SAVE** Armazena em um dispositivo especificado, um programa em BASIC.

Formato: SAVE "nome do dispositivo: nome do arquivo"

**SCREEN** Determina o tipo de tela, o tamanho do "sprite", o "liga/desliga" do clic das teclas, velocidade de transferência de dados para o gravador cassete e seleciona o tipo de impressora.

Formato: SCREEN número de 0 a 3 que indica o tipo de tela, número de 0 a 3 que indica o tamanho do "sprite", número 0 ou 1 que determina ou não o clic do teclado, número 1 ou 2 que indica a velocidade de transferência, número 0 ou 1 que indica o tipo de impressora.

**SNG** Função que verifica o sinal de um número ou expressão.

Formato: SNG (expressão)

**SIN** Função que calcula o valor do seno de um ângulo medido em radianos

Formato: SIN (ângulo)

**SOUND** Escreve dados diretamente nos registros do gerador de sons programável (PSG).

Formato: SOUND número de registro entre 0 e 13,  
expressão inteira entre 0 e 255

**SPACES\$** Função que permite obter uma string com um número especificado de espaços.

Formato: SPACE\$ (argumento entre 0 e 255)

**SPC** Função que permite obter um certo número de espaços.

Formato: SPC (número de espaços)

**SPRITE** Cada uma dessas três instruções respectivamente, habilita, desabilita e adia uma interrupção por sobreposição de "sprites".

**ON/OFF/STOP**

	Formato: SPRITE ON SPRITE OFF SPRITE STOP
<b>SPRITES</b>	Variável que deve ser utilizada para definir os dados de um "sprite".
	Formato: SPRITES (número de "sprite" compreendendo entre 0 e 255 para 8x8 pontos e entre 0 e 63 para 16x16 pontos) = string
<b>SQR</b>	Função que fornece o valor da raiz quadrada de um número ou expressão.
	Formato: SQR (expressão não negativa)
<b>STICK</b>	Função que mostra o estado dos "joysticks" e das teclas que movem o cursor.
	Formato: STICK (n) onde, n = 0 (teclado) n = 1 (joystick A) n = 2 (joystick B)
<b>STOP</b>	Interrompe a execução do programa.
	Formato: STOP
<b>STOP ON/ OFF/STOP</b>	Cada uma dessas três instruções respectivamente, habilita, desabilita e adia uma interrupção por pressão das teclas CTRL+STOP.
	Formato: STOP ON STOP OFF STOP STOP
<b>STRIG</b>	Função que permite obter o estado de toque da barra de espaços e dos botões disparadores dos "joysticks".
	Formato: STRIG (número entre 0 e 4)
<b>STRIG (n) ON/OFF/ STOP</b>	Cada uma dessas três instruções respectivamente, habilita, desabilita e adia uma interrupção por pressão da barras de espaços ou dos botões disparadores dos "joysticks".
	Formato: STRIG (n) ON STRIG (n) OFF (n = 0 a 4) STRIG (n) STOP
<b>STR\$</b>	Função que transforma dados numéricos em alfanuméricicos (strings).
	Formato: STR\$ (expressão numérica)
<b>STRINGS</b>	Função que gera uma string definida como outra string ou outro caracter multiplicado por um certo número de vezes.

	Formato: STRING\$ (número de vezes que o caractere ou a string serão repetidos, código do caractere ou string a ser repetida)
<b>SWAP</b>	Troca o valor de duas variáveis.
	Formato: SWAP variável A, variável B
<b>TAB</b>	Função que tabula a posição do cursor de um certo número de espaços contados desde a margem esquerda.
	Formato: TAB (número de espaços entre 0 e 255)
<b>TAN</b>	Função que calcula o valor da tangente de um ângulo medido em radianos.
	Formato: TAN (ângulo diferente de PI/2 + k*PI)
<b>TIME</b>	Variável que a cada 1/60 segundos é incrementada em uma unidade.
	Formato: TIME TIME = expressão
<b>TROFF</b>	Desliga o funcionamento do comando TRON.
	Formato: TROFF
<b>TRON</b>	Permite visualizar na tela o número da linha do programa que está sendo executada.
	Formato: TRON
<b>USR</b>	Função que executa e fornece o resultado obtido pela execução de uma rotina em linguagem de máquina, que começa no endereço definido por DEFUSR.
	Formato: USR número da rotina entre 0 e 9, (valor a ser transferido para a rotina em LM)
<b>VAL</b>	Função que transforma dados alfanuméricos em numéricos.
	Formato: VAL (string ou variável alfanumérica)
<b>VARPTR</b>	Função que permite obter o endereço do byte a partir do qual está armazenado o conteúdo de uma variável.
	Formato: VARPTR (variável)
<b>VDP</b>	Lê e insere dados diretamente nos registros do processador de vídeo.
	Formato: VDP (número do registro entre 0 e 8) VDP (número do registro) = expressão inteira entre 0 e 255
<b>VPEEK</b>	Lê o byte armazenado no endereço especificado na

	RAM de vídeo.
VPOKE	Formato: VPEEK (endereço entre 0 e 16383) Insere dados num endereço da VRAM (RAM de vídeo).
	Formato: VPOKE endereço entre 0 e 16383, dado entre 0 e 255
WAIT	Faz uma pausa até que um certo valor surja na porta de entrada/saída especificada. Formato: WAIT número da porta de E/S, expressão 1, expressão 2
WIDTH	Determina o número de caracteres por linha no modo texto (SCREEN 0 ou 1). Formato: WIDTH número de caracteres

Para finalizar este capítulo é apresentado um programa em HOT BASIC.

Ele apresenta na tela uma lista de todas as palavras reservadas do HOT BASIC.

Figura 4.49 — Palavras reservadas.

```

100 REM Palavras Reservadas do Hot-Basic
110 WIDTH 38:KEYOFF:SCREEN 0
120 EN=14962
130 I=65:L=0:C=0
140 PRINT CHR$(I);
150 P=PEEK(EN)
160 P$=CHR$(P)
170 IF P<128 THEN PRINT P$:=GOTO 290
180 PRINT CHR$(P-128)
190 L=L+1
200 IF L<=20 THEN 260
210 L=0:C=C+10
220 IF C>=30 THEN 260
230 C=0:PRINT
240 PRINT" Aperte uma tecla ...";
250 PA%=input$(1):CLS
260 LOCATE C,L
270 EN=EN+1
280 IF PEEK(EN+1)<>0 THEN PRINT CHR$(I);
290 IF PEEK(EN)<>0 THEN 330
300 I=I+1:Q$=CHR$(I)
310 IF Q$="J" OR Q$="Q" THEN 330
320 PRINT Q$;
330 EN=EN+1
340 IF EN<=15649 THEN 150
350 LOCATE,21:END

```

# HOT DATA - Gerenciador de Dados

O HOT DATA é um programa gerenciador de banco de dados feito especialmente para o seu HOTBIT e fornecido em cartuchos.

Assim que é instalado, o HOT DATA transforma seu micro-computador num poderoso fichário inteligente onde qualquer tipo de informação pode ser armazenada de forma organizada.

Sua operação é facilíma e ele ainda dispõe de muitas outras vantagens sobre os fichários tradicionais.

Os dados são introduzidos através do teclado e vão sendo armazenados na forma de fichas na memória do micro. Terminada a digitação, o fichário assim constituído pode ser impresso (através de uma impressora) ou armazenado em fita cassete ou disquete para ser usado futuramente.

Entre as muitas facilidades no manuseio dos dados armazenados pelo HOT DATA podemos destacar as seguintes:

- \* Definição do formato das fichas de acordo com a aplicação a que o fichário se destina.
- \* Os dados de cada ficha podem ser digitados sem nenhuma preocupação com a disposição. Apenas na hora da impressão é que a estética deve ser definida.
- \* As fichas podem ser preenchidas em qualquer ordem, pois são automaticamente ordenadas pelo HOT DATA de acordo com o critério definido pelo usuário e em tempo bastante reduzido.

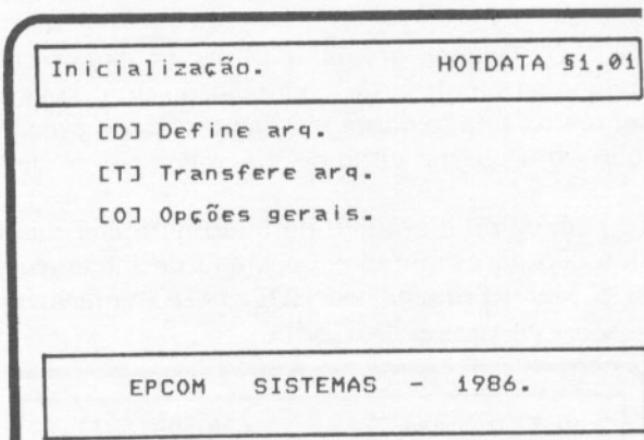
Todas as informações necessárias sobre a instalação e operação do HOT DATA podem ser encontradas no livro "HOT DATA — Gerenciador de dados".

Vamos ver agora, de forma resumida, o essencial para podermos operar o programa. Apesar de ser auto-explicativo e funcionar através de menus bastante claros, é conveniente saber ao menos o essencial sobre ele antes de tentar usá-lo.

O HOT DATA é fornecido num cartucho e pode ser instalado em qualquer um dos dois encaixes (superior ou lateral) do seu HOTBIT. Porém atenção! Durante essa operação o micro deve estar desligado.

Assim que começa a ser executado, o programa gera no vídeo uma tela de apresentação. Após alguns segundos, ou mediante o pressionamento de uma tecla, é apresentada uma tela como a mostrada na figura 5.1.

Figura 5.1 — Menu de Inicialização do HOT DATA.



Essa tela apresenta o Menu de Inicialização do HOT DATA. É com suas opções que devemos começar a operá-lo.

Caso queiramos definir um novo fichário, devemos escolher a opção “[D] Define arq.” que nos permite criar um formato de ficha específico.

Por exemplo, pode-se ter uma ficha para um cadastro de clientes com os seguintes campos:

NOME: .....

END.: .....

CID.: .....

CEP: ..... UF: .....

Uma vez definido o formato das fichas do arquivo, pode-se começar a preenchê-lo com os dados. Quando o preenchimento terminar, o fichário poderá ser impresso ou armazenado em fita ou disquete para uso futuro.

O HOT DATA possui alguns formatos de fichas pré-definidos. Eles podem ser analisados selecionando-se a opção “[T] Transfere arq.” do Menu de Inicialização e escolhendo um dos formatos:

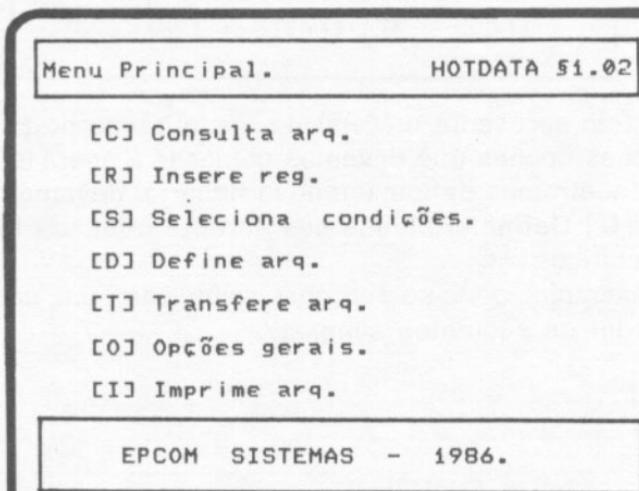
- \* Endereços
- \* Notas
- \* Contatos
- \* Lista de sócios

- \* Arquivo pessoal
- \* Arquivo pessoal
- \* Coleção de selos
- \* Fitas de vídeo

A opção “[T]” também permite que fichários já preenchidos com dados sejam gravados ou carregados de fitas cassette ou disquetes.

A opção “[O] Opções gerais” permite ao usuário alterar ou consultar alguns parâmetros de operação do programa, como as cores presentes na tela, o conteúdo das teclas de funções, informações sobre o fichário definido na memória e dados para impressão.

Quando já existe um arquivo definido na memória do micro, novas opções são acrescentadas ao Menu de Inicialização e ele passa a ser o Menu Principal do HOT DATA. Veja a figura 5.2.  
Figura 5.2 — Menu Principal do HOT DATA.



As novas opções permitem a consulta ao arquivo, o preenchimento de novas fichas, a definição de critérios para a procura e ordenação das fichas durante a consulta ou impressão e a impressão das fichas.

O HOT DATA é um programa residente em cartucho e controla apenas arquivos na memória RAM do micro. Todos os 64 Kbytes do HOTBIT ficam livres para serem preenchidos apenas com dados. Além disso, se você tiver uma expansão de RAM para o micro, o HOT DATA poderá usá-la, aumentando o número de fichas que ele pode controlar simultaneamente.

O tamanho máximo de cada ficha é de 255 caracteres. Isso faz com que os menores arquivos do HOT DATA tenham por volta de 210 fichas e os maiores arquivos (sem expansão de RAM) tenham cerca de 2565 fichas.

A seguir, apresentamos um resumo dos principais comandos do HOT DATA. Como eles correspondem a opções de seus vários menus, vamos analisar cada menu, tela por tela.

Cada tela de opções possui um número em seu canto superior esquerdo, que a identifica. A seguir, elas estão relacionadas de acordo com esses números. Em cada tela, a escolha de uma das opções apresentadas pode ser feita pressionando-se a tecla correspondente à letra apresentada entre colchetes. Pode-se, também, levar o cursor sobre a opção desejada e, a seguir, pressionar **RETURN**.

## RELAÇÃO DAS TELAS DO HOT DATA

### 1.1 — Tela de Abertura.

Tela de apresentação do HOT DATA. Aparece logo depois que o programa é transferido para a memória do HOTBIT. Pressionando-se qualquer tecla ou aguardando-se alguns segundos, será apresentado o Menu de Inicialização.

### 1.01 — Inicialização.

Apresenta as opções iniciais do HOT DATA:

- [D] Permite definir um novo fichário.
- [T] Transfere para a memória um fichário armazenado anteriormente na fita ou no disco.
- [O] Permite alterar os parâmetros gerais.

### 1.02 — Menu Principal.

O Menu Principal é apresentado somente quando o usuário já definiu o fichário de trabalho. Esse menu permite escolher uma das opções seguintes:

- [C] Consulta as fichas individualmente. Esta opção é disponível somente quando o fichário possuir pelo menos 1 ficha preenchida.
- [R] Acrescenta novas fichas.
- [S] Permite definir as condições para a seleção das fichas que serão apresentadas ou impressas.
- [D] Altera o nome ou o tamanho dos campos de um fichário. Atenção para as seguintes restrições:

- 1) O tamanho do campo só pode ser alterado enquanto não houver nenhuma ficha preenchida.
  - 2) O nome do campo pode ser alterado mesmo que o fichário já contenha fichas preenchidas.
  - 3) Novos campos podem ser incluidos, mesmo que o fichário já contenha fichas preenchidas.
- [T] Transfere o fichário para um arquivo em fita cassete ou limpa toda a memória para podemos trabalhar com outro fichário.
- [O] Permite alterar os parâmetros gerais.
- [I] Imprime os dados das fichas.

## 2.01 — Consulta arq.

Essa tela apresenta todos os dados das fichas. Suas opções são:

- [+] Folheia para frente, ficha por ficha. Embora a opção esteja indicada pela tecla + , não há necessidade de se pressionar a tecla SHIFT.
- [–] Folheia para trás, ficha por ficha.
- [F] Imprime os dados da ficha que está sendo apresentada na tela.
- [R] Insere novas fichas.
- [E] Edita a ficha que está sendo apresentada na tela, para correções dos dados.
- [R] Insere novas fichas.
- [C] Cancela a ficha, isto é, elimina-a do fichário.
- [ESC] Volta ao Menu Principal.

## 2.02 — Insere reg.

Apresenta uma ficha em branco para ser preenchida. O cursor estará posicionado no campo de classificação das fichas. Pressione-se a tecla **RETURN** após o preenchimento do campo chave, o HOT DATA irá efetuar uma pesquisa em todas as fichas existentes. Não encontrando qualquer ficha com o mesmo dado, será apresentada a tela 2.04 com os demais campos da ficha, para que o preenchimento possa ser completado. Encontrando alguma ficha que tenha o mesmo dado, será apresentada a tela 2.3 para que o usuário possa desistir da inclusão, insistir na inclusão ou alterar algum dado da ficha existente.

## 2.03 — Formato atual.

Apresenta os dados da ficha existente que tem o mesmo campo chave da nova ficha que está sendo inserida. O usuário poderá escolher uma das opções seguintes:

- [C] Insiste na inclusão da nova ficha.
- [E] Edita a ficha existente e corrige eventuais erros de digitação.
- [ESC] Volta ao Menu de Consulta.

## 2.04 — Novo reg.

Apresenta os demais campos da ficha para que o preenchimento possa ser completado. Durante o preenchimento dos campos, as teclas abaixo possuem as seguintes funções:

▲	Desloca o cursor para o início da linha de cima. Quando o cursor estiver na primeira linha da tela o deslocamento será para a última linha.
▼	Desloca o cursor para o início da linha de baixo. Quando o cursor estiver na última linha da tela, o deslocamento será para a primeira linha.
◀	Desloca o cursor para o caracter à esquerda. Quando o cursor estiver na primeira coluna da linha, não haverá deslocamento.
▶	Desloca o cursor para o caracter à direita. Quando o cursor estiver na última coluna da linha, não haverá deslocamento.
HOME	Desloca o cursor para o início da linha.
DEL	Elimina o caracter sob o cursor e desloca para a esquerda aquilo que estava à sua direita.
INS	Insere um espaço em branco, deslocando para a direita o que estiver à direita do cursor.
◀◀	Desloca o cursor para a esquerda apagando os caracteres sobre os quais ele passar. O que estiver à direita do cursor também será deslocado.
SHIFT CLS	Apaga toda a linha onde se encontra o cursor.
RETURN	Desloca o cursor para a linha de baixo. Estando o cursor na última linha da tela, será apresentada a tela 2.05 para que seja confirmada ou não a inclusão.

**ESC** Apresenta a tela 2.05 para que seja confirmada ou não a inclusão.

## 2.05 — Verifica reg.

Terminado o preenchimento, o HOTDATA irá solicitar que o usuário confirme a inclusão. O usuário deve conferir os dados digitados e pressionar a tecla **S** para que a inclusão seja efetivada. Querendo desistir da inclusão da ficha que foi preenchida, basta pressionar a tecla **ESC**.

## 2.06 — Edita reg.

Essa tela permite que os dados de uma ficha sejam alterados. Para alterar ou corrigir algum dado da ficha, utilize as teclas de movimentação do cursor e de edição conforme apresentado no item 2.04.

## 2.07 — Cancela.

Onde é possível confirmar ou desistir do cancelamento da ficha que está sendo apresentada na tela.

## 3.01 — Define arq.

Onde é possível escolher uma das opções seguintes:

- [N] Permite definir os nomes dos campos.
- [N] Permite alterar os nomes dos campos.
- [N] Inclui mais campos na ficha, mesmo que o fichário já tenha algumas fichas preenchidas.
- [R] Permite definir as restrições do tipo de dado e tamanho máximo da informação que poderá ser datilografada no campo. Essa opção só é possível, enquanto o fichário não tiver nenhuma ficha preenchida.
- [C] Permite definir o campo, pelo qual as fichas ficarão ordenadas no fichário.
- [ESC] Volta ao Menu Principal.
- [ESC] Volta ao Menu de Inicialização, caso não tenha sido definido nenhum novo fichário.

### 3.02 — Nome dos campos.

Essa tela permite definir os nomes de todos os campos que farão parte da ficha. O cursor poderá ser deslocado para qualquer posição dos campos com o auxílio das teclas: ▲, ▼, ◀, ▶ e **HOME**, e os dados poderão ser editados, isto é, corrigidos com auxílio das teclas: **INS**, **DEL**, ◀◀, e **CLS**. O sinal > na frente de um dos campos indica o campo chave, isto é, o campo segundo o qual as fichas ficarão arquivadas.

### 3.03 — Entra restrições.

Essa tela permite definir o tipo do campo e o tamanho máximo das informações que poderão ser datilografadas nos campos da ficha. O cursor poderá ser deslocado para qualquer posição dos campos com o auxílio das teclas: ▲, ▼, ◀, ▶ e **HOME**, e os dados poderão ser editados, isto é, corrigidos e alterados com o auxílio das teclas: **INS**, **DEL** ◀◀ e **CLS**. Só será possível acessar essa opção enquanto o Banco de Dados não tiver nenhuma ficha preenchida.

### 3.04 — Campo chave.

Permite escolher o campo pelo qual as fichas ficarão ordenadas no fichário. Qualquer um dos campos da ficha pode ser escolhido como campo chave. As fichas poderão ser ordenadas e re-ordenadas a qualquer instante, bastando escolher o novo campo chave de classificação. Nas consultas pelo vídeo e nas relações impressas, as fichas serão apresentadas na mesma ordem em que elas estiverem classificadas.

### 3.05 — Classificando.

Essa tela é apresentada durante a classificação das fichas. Ao final da classificação será apresentada a quantidade total de fichas classificadas.

### 4.01 — Lê arquivo.

Essa tela só aparecerá quando existir um drive conectado ao HOTBIT. Caso este drive não exista aparecerá a tela 4.02.

[L] Permite transferir um arquivo gravado em disquete para a memória do HOTBIT.

[C] Transfere um fichário armazenado na fita casse-

te para a memória do HOTBIT.

[P] Permite escolher um dos fichários pré-definidos no HOT DATA.

[ESC] Volta ao Menu Principal.

#### 4.02 — Lê arquivo.

Essa opção só é possível se a memória estiver limpa, isto é, não tiver nenhum fichário definido. Com ela é possível escolher uma das seguintes opções:

[S] Permite escolher um dos fichários pré-definidos no HOT DATA.

[C] Transfere um fichário armazenado na fita cassette para a memória do HOTBIT.

[ESC] Volta ao Menu Principal.

#### 4.03 — Exemplos.

Essa opção só é possível se a memória estiver limpa. Com ela é possível escolher um dos fichários pré-definidos:

##### [A] Endereços.

Nome	24
Iniciais	6
N. casa	6
Av/rua	24
Cidade	24
Cep	C7
Fone	15
Condição	1

##### [B] Notas.

Data	D
Dia	3
Hora	+5
Lugar	24
Com	24
Obs	48
N. da semana	#2

##### [C] Contatos.

Contato	24
Empresa	24
Endereço	20
Cidade	24
CEP	C7
Fone	15
Tipo	16
Visita	D
Categoria	2
Obs.	24

[D] Lista de sócios.		[E] Arq pessoal.	
Nome	24	Nome	24
Iniciais	8	Iniciais	24
1. nome	12	Endereço	24
Endereço	24	Cidade	24
Cidade	24	CEP	C7
CEP.	C7	Fone	15
Fone	15	Nascido a:	D
Registrado a	D	Cargo	18
Manutenção	\$6	Instrução	24
Sócio n.	#4	Admissão	D
Nascido a:	D	Est. civil	C1
Sexo (M/F)	C1	Sexo (M/F)	C1
Est civil.	C1	Filhos	1
Obs.	24	Salário	\$8

[F] Colecão de selos.		[G] Fitas vídeo.	
País	20	Nome filme	20
Catálogo	#8	Autor	20
Ano	#4	Categoria	20
Valor selo	5	Data rec.	D
Data Selo	D	Fita n.	#4
Tipo	10	Tempo	4
Franquia S/N	1	Contador	#4
Val compra	\$7		
Valor atual	\$7		
Número	#3		
Obs	24		

#### 4.04 — Lendo cassete.

Essa tela é apresentada durante a transferência da fita para a memória. O nome do arquivo, inicialmente em branco, será preenchido, tão logo seja localizado, na fita, o arquivo solicitado. O número de registros apresenta, inicialmente, o número zero. A medida em que os registros vão sendo transferidos para a memória, esse número vai sendo incrementado, retratando a situação da transferência.

#### 4.11 — Salva arquivo.

Onde é possível escolher uma das seguintes opções:

- [V] Altera a condição de verificação. Com esta opção ativada, serão transferidas para a fita cassete, somente as fichas que satisfaçam as condições estabelecidas no item 5.01.
- [W] Limpa a memória para podermos trabalhar com um outro fichário. Antes de usá-la, porém, o usuário deve salvar o fichário que está na memória, transferindo-o para uma fita cassete.
- [C] Armazena o fichário em uma fita cassete.
- [ESC] Volta ao Menu Principal.

#### 4.12 — Salva arquivo.

- [V] Altera a condição de verificação. Com esta opção ativada, serão transferidas para a fita cassete, somente as fichas que satisfaçam as condições estabelecidas no item 5.01.
- [S] Transfere um fichário da memória do HOTBIT para um disquete.
- [C] Armazena o fichário em uma fita cassete.
- [M] Limpa a memória para podermos trabalhar com um outro fichário. Antes de usá-la, porém, o usuário deve salvar o fichário que está na memória, transferindo-o para uma fita cassete.
- [ESC] Volta ao Menu Principal.

#### 4.13 — Limpa toda memória.

Onde é possível escolher uma das seguintes opções:

- [S] Confirma a opção de limpeza da memória para podemos trabalhar com um outro fichário.
- [ESC] Volta ao menu de Salva arquivo.

#### 4.14 — Salvando em cassete.

Essa tela será apresentada durante a transferência do fichário para a fita cassete.

#### 4.15 — Verifica tape.

Essa tela permite que seja verificado o que foi gravado na fita. O usuário deve voltar a fita para o início do arquivo. O HOT DATA irá ler o que foi gravado e comparar com o que tem na memória.

#### 4.30 — Erro/Transf.

Essa tela é apresentada em situações onde ocorrer algum erro durante a transferência do arquivo.

### 5.01 — Condições.

Essa tela permite fixar as condições segundo a qual as fichas devem ser apresentadas ou impressas. Uma condição é estabelecida fixando os seus limites inferior e superior. A tela permite escolher:

- [+] Fixa o limite inferior da condição.
- [–] Fixa o limite superior da condição.
- [P] Promove a procura das fichas que satisfaçam a condição estabelecida.
- [ESC] Volta ao Menu Principal.

#### 5.02 — Maior ou igual.

Essa tela permite fixar o limite inferior da condição de procura. Escolha o campo de seleção, deslocando o cursor até ele com o auxílio das teclas ▲ e ▼. Digita-se o limite inferior da condição e pressiona-se a tecla **RETURN**. Em função das necessidades do usuário, vários campos podem conter condições de seleção.

#### 5.03 — Menor ou igual.

Essa tela permite fixar o limite superior da condição de procura. Indique o limite superior de maneira análoga à descrita no item anterior.

#### 5.04 — Seleciona condição.

Essa tela permite consultar individualmente as fichas que atendem as condições estabelecidas no item 5.01. As opções

apresentadas por essa tela são as seguintes:

- [+] Folheia para a frente, procurando outra ficha que satisfaça a condição.
- [–] Folheia para trás, procurando outra ficha que satisfaça a condição.
- [F] Imprime os dados da ficha que está sendo apresentada na tela.
- [C] Cancela, isto é, elimina do arquivo, a ficha que está sendo apresentada na tela.
- [ESC] Volta ao Menu Principal.

#### 6.01 — Impressão.

Onde é possível escolher uma das seguintes opções:

- [L] Imprime uma relação na forma de tabela.
- [F] Imprime uma relação na forma de etiquetas.
- [V] Indica se na listagem a condição de seleção deve ou não ser seguida. Estando ativada, somente as fichas que satisfizerem a condição estabelecida, serão impressas.

#### 6.11 — Lista.

Onde é possível escolher uma das seguintes opções:

- [I] Imprime a relação.
- [D] Permite definir os campos que farão parte da relação.
- [P] Permite a escolha de formulário contínuo ou folhas soltas.
- [C] Permite fixar o comprimento do papel.
- [N] Permite fixar a quantidade de linhas que deverão ser impressas por folha.
- [T] Permite definir um título que será impresso no topo de todas as folhas.
- [ESC] Volta ao Menu de Impressão.

#### 6.12 — Imprime lista.

Essa tela é apresentada durante a impressão. Desejando-se interromper a impressão, basta pressionar as teclas [CTRL]+[STOP].

## 6.13 — Define lista.

Essa tela permite escolher os campos que serão impressos. A tela apresenta 3 colunas. Na primeira coluna estão relacionados todos os campos da ficha. Na segunda coluna, é apresentado um espaço de 14 caracteres onde poderá ser digitado o título do campo. Na terceira coluna deverá ser digitado o número da coluna a partir da qual o campo deve ser impresso. As opções dessa tela são as seguintes:

- [T] Permite definir os títulos dos campos.
- [P] Permite definir a posição do campo na linha impressa.
- [ESC] Permite voltar ao Menu de Impressão.

## 6.14 — Texto.

Permite definir os títulos dos campos que serão impressos no cabeçalho da página. Para digitar, alterar e corrigir os títulos, podem ser usadas as teclas de edição: ▼, ▲, ◀, ▶, INS, DEL, ◀◀, HOME e CLS.

## 6.15 — Posição Horizontal.

Permite fixar a coluna a partir da qual o campo deverá ser impresso.

## 6.21 — Form.

Onde é possível escolher uma das seguintes opções:

- [I] Imprime a relação.
- [D] Permite definir os campos que farão parte da relação.
- [P] Permite a escolha de formulários contínuos ou folhas soltas. Quando ativada, faz com que a impressão seja feita de forma contínua, sem interrupções. Quando desativada, a impressão é interrompida ao final de cada página. É muito útil quando se trabalha com folhas soltas.
- [C] Permite fixar o comprimento do papel ou etiqueta. No caso de etiquetas autocolantes, medir o espaçamento entre uma etiqueta e outra e colocar aqui a quantidade de linhas que cabem entre o topo de uma etiqueta e o topo da se-

guinte. Lembrar que a impressão normal é com um espaçamento vertical correspondente a 6 linhas por polegada.

- [E] Permite definir um texto que será impresso em todas as etiquetas.
- [T] Permite indicar a posição em que o texto deverá ser impresso.
- [ESC] Volta ao Menu de Impressão.

#### 6.22 — Imprime form.

Essa tela é apresentada durante a impressão. Desejando-se interromper a impressão, basta pressionar as teclas: [CTRL]+[STOP].

#### 6.23 — Define form.

Essa tela permite escolher os campos que serão impressos. A tela apresenta três colunas. Na primeira estão relacionados todos os campos da ficha. Na segunda, deverá ser digitado o número da linha onde o campo aparecerá. Na terceira deverá ser digitado o número da coluna, a partir da qual o campo será impresso. As opções desta tela são:

- [V] Permite indicar o número da linha.
- [H] Permite indicar o número da coluna.
- [ESC] Volta ao Menu de Impressão.

#### 6.24 — Posição Vertical.

A posição vertical é o número da linha onde o usuário deseja que o campo seja impresso.

#### 6.25 — Posição Horizontal.

A posição horizontal é o número da coluna onde o usuário deseja que o campo seja impresso.

#### 7.01 — Opções gerais.

Onde é possível escolher uma das opções seguintes:

- [C] Permite escolher uma outra cor para as letras.

- [B] Permite escolher uma outra cor para o fundo.
  - [M] Permite indicar se a impressora aceita ou não os caracteres acentuados do padrão BRASCII.
  - [L] Permite indicar se a impressora necessita ou não do caracter Line-feed após o caracter CTRL+M.
  - [T] Permite indicar se a impressora será alimentada com formulário contínuo ou folhas soltas.
  - [F] Permite atribuir nomes às teclas de funções.
  - [I] Obtém informações sobre a estrutura do Banco de Dados carregado na memória.
- [**ESC**]. Volta ao Menu Principal.

#### 7.02 — Cor da letra.

Nessa tela, o usuário poderá escolher a cor das letras. Para isso, pressione a tecla com a letra entre colchetes. Desejando-se manter a cor que está em uso, pressione **ESC**.

#### 7.03 — Cor de fundo.

Nessa tela, o usuário poderá escolher a cor de fundo. Para isso, pressione a tecla com a letra entre colchetes. Desejando-se manter a cor que está em uso, pressione **ESC**.

#### 7.04 — Teclas de função.

Essa tela permite que sejam atribuidas palavras e frases para cada uma das teclas de funções do HOTBIT, para facilitar o trabalho de preenchimento das fichas. Informações repetitivas, tais como: o nome da Cidade, podem ser atribuidas a uma das teclas de função para que durante a datilografia das fichas, tais nomes sejam preenchidos sem que se precise datilografá-los, bastando pressionar a tecla de função correspondente. O comprimento máximo da frase deverá ser de 15 caracteres.

#### 7.05 — Configuração arq.

Essa tela apresenta as principais características do fichário montado na memória, informando o seguinte:

- Memória útil  
A quantidade total de bytes disponíveis para a montagem do Banco de Dados.
- N. registros  
A quantidade total de registros (ou fichas) que cabem no Banco de Dados.
- N. ocupados  
A quantidade de registros preenchidos.
- Tamanho reg  
A quantidade de bytes utilizados por registro do Banco de Dados.
- Nome arq  
O nome do arquivo no qual o Banco de Dados será armazenado. Pressionando-se a tecla **RETURN** ou **ESC** o HOT DATA apresentará o Menu das Opções Gerais.

# HOT PLAN - Planilha de Cálculos

O HOT PLAN é um programa de planilha de cálculos fornecido em cartuchos e feito especialmente para o HOTBIT.

Quando instalado, o HOT PLAN transforma seu HOTBIT numa poderosíssima planilha eletrônica onde centenas de cálculos poderão ser feitos automaticamente e num tempo curtíssimo.

Rapidamente veremos o que é e para que serve uma planilha eletrônica. E depois como funciona o programa.

Algumas atividades exigem um controle freqüente e envolvem grande número de itens a serem verificados. Muitas vezes esses itens estão intimamente relacionados entre si e é nesses casos que o uso de uma planilha de cálculos se torna indispensável.

Imagine uma enorme tabela com 31 linhas e 12 colunas, onde estão anotados valores diários relativos a um determinado investimento, sendo que todos os valores estão expressos numa unidade padrão (dólares, por exemplo), mutável de dia para dia!

Esse é um exemplo bem simples, mas mesmo assim, imagine só o trabalho que daria ter que alterar toda a tabela cada vez que o dólar mudasse de valor!

Numa planilha eletrônica, como o HOTPLAN, esse trabalho é incrivelmente simplificado. Basta alterar uma única posição da tabela para que todos os valores sejam automaticamente corrigidos!

Existem muitas outras aplicações mais complexas onde o uso de uma planilha eletrônica se faz indispensável.

Apresenta-se, a seguir, uma relação completa de todos os comandos, funções e teclas, em ordem alfabética, do HOT PLAN, a planilha eletrônica do seu HOTBIT.

## RESUMO DE COMANDOS, FUNÇÕES E TECLAS

**ABS(argumento)** Retorna o valor absoluto do argumento.

Exemplo: ABS(R7C4)

**AND(lista)** Examina as várias condições contidas numa lista e retorna à condição verdadeira, se todas elas forem verdadeiras. Exemplo:

**IF(AND(Soma>2000,Vontade,Dívidas<500),R12C5,R13C5)**

**ATAN(argumento)** Calcula o arco tangente em radianos do argumento. Exemplo:

ATAN(R3C2/R4C3)

**AVERAGE(lista)** Calcula a média aritmética da lista.

Exemplo: AVERAGE(Menores.que.12)

**CALL HOTPLAN** Inicializa o HOT PLAN, logo após ter sido carregado na memória.

**CALL HOTPLANP** Inicializa o HOT PLAN sem as máscaras de linhas e de colunas.

**Célula** É a interseção de uma linha com uma coluna.

As células são referenciadas por meio das letras R e C seguidas de números. Exemplos:

R27C5 Célula formada pela interseção da linha 27 com a coluna 5.

R9:15C3 Conjunto de elementos da coluna 3, a partir da linha 9 até a linha 15.

R10C2:12 Conjunto de elementos da linha 10, a partir da coluna 2 até a coluna 12.

R10:30C Conjunto dos elementos da mesma coluna onde está a fórmula, desde a célula da linha 10 até a célula da linha 30.

R12C[-1] Célula situada na interseção da linha 12 com a coluna da esquerda de onde está a fórmula.

R[+1]C Célula situada logo abaixo da célula onde está a fórmula.

R[1]C O mesmo que R[+1]C

R[-5]C:R[-1]C Conjunto das células na mesma coluna, a partir do quinto elemento à esquerda até o primeiro elemento acima da célula onde está a fórmula.

**COLUMN( )** Retorna o número da coluna.

Exemplo: COLUMN( )+100

**COS(argumento)** Calcula o co-seno do argumento em radianos.

Exemplo: COS(R3C2/PI( )/360)

**COUNT(lista)** Retorna a quantidade de elementos da lista.

Exemplo: COUNT(R3:45C2:7)

**CTRL A** Comando de Edição, igual a F1.

<b>CTRL B</b>	Comando para limpar células, igual a F2
<b>CTRL C</b>	Comando para copiar células, igual a F3
<b>CTRL D</b>	Comando para fixar o formato das células, igual a F4.
<b>CTRL E</b>	Comando para deslocar diretamente o cursor, igual a F5.
<b>CTRL G</b>	Transforma a referência relativa, da posição do cursor, em referência absoluta.
<b>CTRL H</b>	Igual a tecla <<, ou Backspace.
<b>CTRL K</b>	Desloca o cursor para a primeira célula, igual a <b>HOME</b>
<b>CTRL M</b>	Igual a tecla <b>RETURN</b> .
<b>CTRL R</b>	Comando para inserir novas linhas ou colunas, igual a <b>INS</b> .
<b>CTRL S</b>	Comando para atribuir um nome, igual a <b>F6</b> .
<b>CTRL T</b>	Comando para fixar opção de cálculo, igual a <b>F7</b> .
<b>CTRL U</b>	Comando para imprimir a planilha, igual a <b>F8</b> .
<b>CTRL V</b>	Comando para transferir a planilha, igual a <b>F9</b> .
<b>CTRL W</b>	Comando para sair do HOT PLAN, igual a <b>F10</b> .
<b>CTRL Z</b>	Desloca o cursor para o fim da planilha.
<b>DEL</b>	Deleta linhas ou colunas. Permite eliminar as linhas ou colunas indesejáveis da planilha.

**EXP(argumento)** Calcula o número e (número de Euler=2.71...) elevado ao argumento.

Exemplo: EXP(R3C2)

**FALSE( )** Retorna o valor lógico Falso.

Exemplo: IF(Verba(20000,"Viável",FALSE( ))

**F1 a F10** As teclas de funções do teclado do HOTBIT representam as opções de trabalho do HOTPLAN. Tais comandos são os seguintes:

- F1** — Comando de Edição
- F2** — Comando para limpar células
- F3** — Comando para copiar células
- F4** — Comando para fixar o formato da célula
- F5** — Comando para deslocar o cursor
- F6** — Comando para definir nomes de células
- F7** — Comando para fixar o método de cálculo
- F8** — Comando para Impressão
- F9** — Comando para transferência de arquivo
- F10** — Comando para sair do HOT PLAN.

Conforme o comando escolhido, novas opções poderão ser apresentadas.

- HOME** Desloca o cursor para a primeira célula da planilha.
- IF(Condição, Verdadeira, Falsa)** Verifica a validade da Condição. Se for verdadeira, adota a expressão Verdadeira, caso contrário, adota a outra expressão.  
Exemplo: IF(Salário<5000,"Inviável",R5C2)
- INDEX(tabela, m, n)** Retorna o elemento da tabela, situado na m-ésima linha e n-ésima coluna, contados a partir do início da tabela.  
Exemplo: INDEX(R19:25C12:14,5,2)
- INS** Insere linhas e colunas. Permite inserir uma ou várias linhas ou colunas no meio da planilha.
- INT(argumento)** Retorna somente a parte inteira do argumento.  
Exemplo:  
IF(R5C4-INT(R5C4)>=0.5,INT(R5C4)+1,INT(R5C4)+0.5)
- ISERROR(elemento)** Retorna o valor lógico Verdadeiro se o elemento contém um dos valores lógicos de erro.  
Exemplo: IF(ISERROR(R4C5),"Verificar","Prosseguir")
- ISNA(elemento)** Retorna o valor lógico Verdadeiro se o elemento contém o valor lógico # N/D.  
Exemplo:  
IF(ISNA(SUM(R12:28C4)),"Faltam Dados",SUM(R12:28C4))
- LN(argumento)** Calcula o logaritmo natural do argumento.  
Exemplo: LN(SUM(Membros))
- LOG10(argumento)** Calcula o logaritmo decimal do argumento.  
Exemplo: LOG10(R4C2+R7C2)
- LOOKUP(n,tabela)** Procura o valor N na primeira coluna da tabela, retornando o valor da última coluna da tabela que corresponde a N. Procura o valor N na primeira linha da tabela, retornando o valor da última linha da tabela que corresponde a N.  
Exemplo: LOOKUP(23.45,C5:12F10:13)
- MAX(lista)** Retorna o maior valor da lista.  
Exemplo: MAX(R16:34C7)
- MIN(lista)** Retorna o menor valor da lista.  
Exemplo: MIN(2340,R3C3,Menores.que.12)
- MOD(dividendo, divisor)** Retorna o resto da divisão do dividendo pelo divisor.  
Exemplo: MOD(R5C3+SUM(R7:12C3),4)
- NA()** Retorna o valor lógico # N/D.  
Exemplo: IF(R12C4<2000,NA(),R10C3)

**NOT(argumento)** Retorna o contrário da condição lógica do argumento.

Exemplo: IF(NOT(R3C2>40),2000,R12C7)

**NPV(taxa, lista)** Calcula o valor presente de um fluxo de caixa representado pela lista, na taxa dada.

Exemplo: NPV(0.025,R3:15C7)

**Números** Os números são armazenados na memória com a precisão de 14 dígitos. Os números reais podem ser armazenados com expoente, variando desde -63 a + 63.

**OR(lista)** Examina as várias condições contidas na lista e retorna à condição Verdadeira, se qualquer delas resultar numa condição verdadeira.

Exemplo: IF(OR(R3C4>100,R4C5>=450),SUM(R12:18C),  
AVERAGE(R12:18C[1]))

**PI( )** Retorna o valor da constante PI.

Exemplo: SIN(PI( )\*2)

**ROUND(Argumento,n)** Arredonda o argumento na n-ésima casa decimal.

Exemplo: ROUND(R4C2=R5C2/2,-3)

**ROW( )** Retorna o número da linha.

Exemplo: ROW( )+1985

**SHIFT ▶** Desloca o cursor para a página da direita.

**SHIFT ◀** Desloca o cursor para a página da esquerda.

**SHIFT ▼** Desloca o cursor para a página de baixo.

**SHIFT ▲** Desloca o cursor para a página de cima.

**SIGN(argumento)** Retorna o valor -1, 0 ou 1, em função do sinal do argumento.

Exemplo: SIGN(R4C3)

**SIN(argumento)** Calcula o seno do argumento em radianos.

Exemplo: SIN(SUM(R4:7C4)+R7C5)

**SQRT(argumento)** Calcula a raiz quadrada do argumento.

Exemplo: IF(SQRT(R14C6)<12,R13C15,R14C15)

**STDEV(lista)** Calcula o desvio padrão da lista.

Exemplo: STDEV(R12:18C4)

**SUM(lista)** Calcula a soma dos elementos da lista.

Exemplo: SUM(R2:45C3)

**TAB** Em comandos que apresentam 2 campos de digitação, esta tecla permite deslocar o cursor de um campo para outro.

**TAN(argumento)** Calcula a tangente do argumento em radianos.

Exemplo: TAN(PI( )\*2)

**Textos** Os textos, contendo fórmulas podem ter até 150 caracteres de comprimento.

**TRUE( )** Retorna o valor lógico Verdadeiro.

Exemplo: IF(Verba>2000,TRUE( ),"Inviável")

► Desloca o cursor para a direita.

No modo Planilha, desloca o cursor para a coluna da esquerda.

No modo de Edição, desloca o cursor para o caracter da direita.

◀ Desloca o cursor para a esquerda.

No modo de Edição, desloca o cursor para o caracter da esquerda.

◀◀ No modo de edição, apaga o caracter à esquerda do cursor, deslocando o resto da linha para a esquerda, inclusive o cursor.

! Comando que leva o HOT PLAN a recalcular todas as fórmulas.

+ Operador algébrico de soma.

- Operador algébrico de subtração.

/ Operador algébrico de divisão.

\* Operador algébrico de multiplicação.

^ Operador algébrico de exponenciação.

% Operador algébrico de porcentagem.

( ) Separador algébrico para fixar a prioridade de cálculos.

: Separador algébrico que indica união de conjuntos.

**SPACE** Separador algébrico que indica intersecção de conjuntos

, Separador de elementos de uma lista.

= Quando colocado no início, indica ao HOT PLAN que a célula contém uma fórmula.

= Operador lógico de igualdade.

> Operador lógico que indica maior que.

>= Operador lógico que indica maior ou igual a.

< Operador lógico que indica menor que.

<= Operador lógico que indica menor ou igual a.

<> Operador lógico que indica diferente de.

# HOT WORD - Processador de Textos

O HOT WORD é um programa processador de textos feito sob medida para o seu HOTBIT e fornecido em cartucho.

Ao ser executado, ele transforma seu microcomputador numa poderosíssima máquina de escrever eletrônica de operação facilíma e ainda com muitas outras vantagens sobre as máquinas tradicionais.

O texto é introduzido através do teclado e vai sendo armazenado na memória do micro. Terminada a digitação, o texto pode ser impresso (através de uma impressora) ou armazenado em fita cassete ou disquete para ser usado futuramente.

Entre as muitas facilidades para manuseio de textos oferecidas pelo HOT WORD podemos destacar as seguintes:

- \* O texto pode ser digitado sem nenhuma preocupação com a sua estética. Apenas na hora da impressão, é que a estética do texto deve ser definida.
- \* Pode-se definir blocos de textos, formados por várias linhas ou vários parágrafos e apagá-los, deslocá-los ou copiá-los em qualquer outra posição do texto.
- \* Pode-se procurar e substituir automaticamente uma palavra por outra em todo o texto. Por exemplo, se ao escrever um contrato, o número do RG de uma pessoa foi digitado errado em todos os lugares em que ele aparece, ao invés de procurar e substituir manualmente cada um deles, pode-se fazer com que o HOT WORD proceda automaticamente a procura e a substituição em todo o texto.
- \* Para a impressão do texto, dependendo do tipo de impressora usada, pode-se dispor de até 48 diferentes caracteres de controle simultaneamente. Isso permite, por exemplo, numa MÔNICA EL 6011, produzir facilmente trechos em negrito, expandido ou condensado, com letra comum ou com "qualidade de carta", etc...

Todas as informações necessárias sobre a instalação e operação do HOT WORD podem ser encontradas no livro "HOT WORD — Processador de texto".

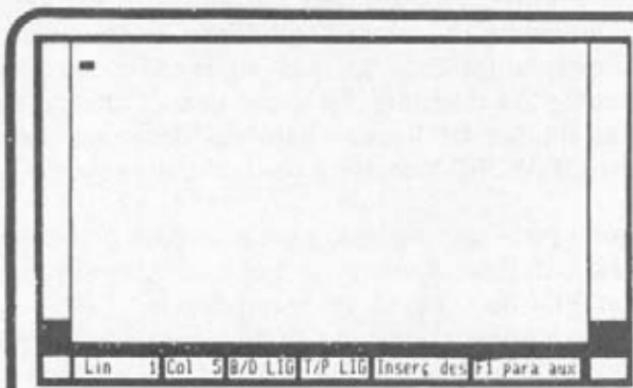
Apenas para termos uma idéia de como ele funciona, va-

mos agora ver alguns detalhes de sua operação.

O HOT WORD é fornecido num cartucho e pode ser instalado em qualquer dos dois encaixes (superior ou lateral) do seu HOTBIT. Cuidado na instalação: certifique-se de que o micro esteja desligado!

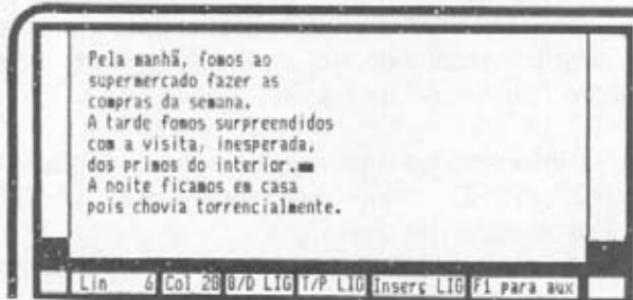
Assim que começa a ser executado o programa apresenta uma tela como a da figura 7.1.

Figura 7.1 — Tela de edição do HOT WORD.



Nessa tela, o texto introduzido pelo teclado vai aparecendo nas 22 primeiras linhas. A posição em que cada caracter a ser digitado aparece, é indicada por um retângulozinho preto (o cursor). Quando toda a tela estiver cheia, a primeira linha desaparecerá (apenas da tela!), todas as outras linhas serão deslocadas para cima e a última linha aparecerá vazia a espera de mais texto. Essa tela se assemelha a um visor que pode ser movido ao longo de um pergaminho. Veja a figura 7.2.

Figura 7.2 — Visor de um pergaminho.



A última linha da tela é usada pelo HOT WORD para apresentar informações úteis ao digitador, como a linha e a coluna em que o próximo caracter a ser digitado aparecerá.

Um recurso muito útil para os digitadores iniciantes é a possibilidade de consulta a duas páginas de auxílio que contém um resumo de comandos do HOT WORD. Para obtê-las no vídeo, basta pressionar a tecla F1. Ao fazer isso será apresentada uma imagem como a da figura 7.3.

Figura 7.3 — Primeira tela de auxílio



Pressionando F1 mais uma vez será apresentada a segunda tela de auxílio. Veja a figura 7.4.

Figura 7.4 — Segunda tela de auxílio.



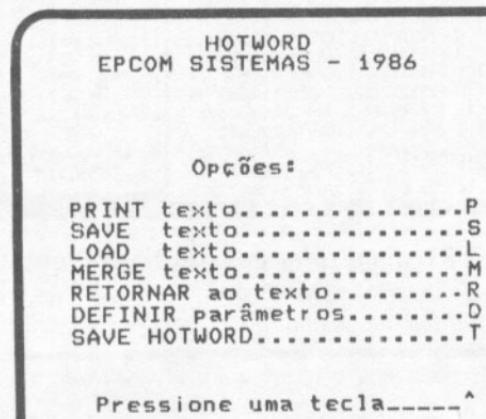
Para retornar ao texto a partir de qualquer das duas telas de auxílio, basta digitar RETURN.

Todas as teclas apresentadas nas telas de auxílio podem ser usadas durante a digitação do texto e correspondem a comandos específicos para o HOT WORD.

Dois comandos são especialmente importantes e podem ser acessados pressionando-se as teclas **CTRL+STOP** ou a tecla **GRAPH**.

Pressionando-se **CTRL+STOP**, aparecerá na tela um menu de serviços do HOT WORD (figura 7.5), no qual pode-se escolher uma entre várias opções para gravação e leitura em disquete ou fita cassete, impressão do texto, alteração de parâmetros de controle do programa, etc...

Figura 7.5 — Menu de serviços.



A tecla **GRAPH** serve para colocarmos no texto seqüências de caracteres de controle para a impressora. Assim, por exemplo, se quisermos imprimir um texto com caracteres condensados numa MÔNICA EL 6011, basta colocar o cursor na primeira posição da primeira linha do texto, digitar a tecla **GRAPH** e digitar a tecla do número 4. Na tela aparecerá o símbolo █ que é interpretado pelo programa como o caracter 15 (controle para condensação na MÔNICA). Os caracteres de controle que existem no HOT WORD são os apresentados na figura 7.6.

Figura 7.6 — Caracteres de controle de impressora.

■ ■	SHIFT 1	■ ■	1
■ ■	SHIFT 2	■ ■	2
■ ■	SHIFT 3	■ ■	3
■ ■	SHIFT 4	■ ■	4
■ ■	SHIFT 5	■ ■	5
■ ■	SHIFT 6	■ ■	6
■ ■	SHIFT 7	■ ■	7
■ ■	SHIFT 8	■ ■	8

As sequências de caracteres podem ser alteradas pelo digitar de acordo com a impressora de que ele dispõe. Para isso, basta escolher a opção **D** (Definir parâmetros) do menu de serviço e proceder as alterações.

Essas são as principais características do programa. A seguir, apresentamos um resumo dos comandos para consultas rápidas.

## COMANDOS PARA A EDIÇÃO DO TEXTO

**CTRL+A** Fixa a Margem Direita.

A margem direita é fixada na coluna de número par anterior a posição do cursor.

**CTRL+B** Marca o início de um bloco de texto.

Uma nova linha em branco é inserida na linha acima de onde está o cursor para conter a marca de início do bloco. O cursor é deslocado para o início dessa linha. O restante do texto é deslocado para baixo.

**CTRL+D** Fixa a Margem Esquerda.

A margem esquerda é fixada na coluna de número ímpar posterior a posição do cursor.

**CTRL+E** Ativa ou Desativa a opção **B/D**.

A opção **B/D**, quando ativada, faz com que o parágrafo que está sendo datilografado vá sendo alinhado em ambas as margens. Quem datilografa, deve

preocupar-se apenas com o conteúdo. A estética deve ser preocupação do Processador de Textos.

- CTRL+F** Desloca a tela para baixo.  
Desloca a tela 22 linhas para baixo, o que equivale a deslocar o texto 22 linhas para cima.
- CTRL+G** Desloca a tela para cima.  
Desloca a tela 22 linhas para cima, o que equivale a deslocar o texto 22 linhas para baixo.
- CTRL+J** Ajusta a Linha.  
Os espaços em branco que existem entre a última letra da linha e a margem direita são distribuídos entre as palavras de forma a alinhar o texto em ambas as margens.
- CTRL+N** Copia o Bloco.  
Uma cópia, exatamente igual ao bloco marcado será criada e inserida entre a linha acima do cursor e a linha onde ele está.
- CTRL+O** Procura uma Palavra.  
Permite fornecer uma determinada palavra para que o HOT WORD a procure no meio do texto. A procura é feita a partir de onde o cursor está.
- CTRL+O** Substitui Palavras.  
Permite fornecer uma palavra ou frase, para que o HOT WORD a substitua, sempre que a encontrar no texto. A substituição é feita a partir de onde o cursor está.
- CTRL+Q** Move o Bloco.  
Desloca o Bloco, anteriormente marcado, para a posição onde está o cursor, inserindo-o entre a linha acima dele e a linha em que ele está.
- CTRL+S** Fixa as Margens no padrão do HOT WORD.  
Se as posições das margens foram alteradas pelos comandos **CTRL+A** e **CTRL+D**, este comando permite voltar aos padrões normais do HOT WORD.
- CTRL+STOP** Apresenta o Menu de Serviço.  
Estando na tela de texto, estas teclas apresentam a tela com o Menu de Serviços.
- CTRL+T** Troca a Janela.  
Troca a forma de apresentação da tela de texto entre o padrão de 32 colunas e o padrão de 64 colunas.

- CTRL+U** Elimina os espaços em branco supérfluos de uma linha, juntando-se todas as palavras e alinhando-as pela margem esquerda.
- CTRL+V** Marca o fim de um bloco.  
Uma nova linha em branco é inserida na linha abaixo do cursor para conter a marca de fim do bloco. O cursor é deslocado para o início dessa linha. O restante do texto é deslocado para baixo.
- CTRL+W** Ativa ou Desativa a opção **T/P**.  
A opção **T/P**, quando ativada, permite que na distribuição automática das palavras de um parágrafo elas sejam mantidas sempre por inteiro, sem ser quebradas.
- CTRL+Y** Apaga todo o texto.  
Apaga todo o texto que esta na memória do HOTBIT, permitindo iniciar a datilografia de um novo texto, sem precisar desligar e ligar de novo o microcomputador.
- CTRL+Z** Ativa ou desativa o modo de inserção.  
Ativa ou desativa o modo de inserção automática de linhas durante a datilografia de um novo parágrafo no meio do texto. O modo de inserção, estando ativado, cria automaticamente as linhas necessárias para a introdução do novo parágrafo.
- CTRL+▲** Texto para cima.  
Desloca o texto uma linha para cima. O cursor é levado junto com o texto. Se o cursor estiver na primeira linha da tela, ele não se moverá, mas o texto será deslocado normalmente.
- CTRL+▼** Texto para baixo.  
Desloca o texto uma linha para baixo. O cursor é levado junto com o texto. Se o cursor estiver na última linha da tela, ele não se moverá, mas o texto será deslocado normalmente.
- CTRL+◀** Palavra à esquerda.  
Desloca o cursor para a última letra da próxima palavra à sua esquerda. Se o cursor estiver na primeira palavra da linha, o deslocamento será para a última palavra da linha de cima. Se o cursor estiver na pri-

meira palavra do parágrafo, o deslocamento será para a última palavra do parágrafo anterior.

**CTRL+►** Palavra à direita.

Desloca o cursor para a primeira letra da próxima palavra à direita. Se o cursor estiver na última palavra da linha, o deslocamento será para a primeira palavra da linha de baixo. Se o cursor estiver na última palavra do parágrafo, o deslocamento será para a primeira coluna da próxima linha em branco.

**DEL** Elimina um espaço em branco ou uma letra.

Elimina o espaço em branco ou a letra sob o cursor. O texto na linha do cursor é deslocado para a esquerda e ele não sai do lugar.

**F1** Páginas de auxílio.

Estando na tela de textos, essa tecla apresenta a primeira página de auxílio.

Estando na primeira página de auxílio, essa tecla apresenta a segunda e vice-versa.

**F2** Reforma o parágrafo.

Com o cursor posicionado na primeira linha de um parágrafo, esse comando redistribui as palavras entre as margens recompondo a estética do parágrafo.

**F3** Elimina uma Linha.

A linha onde está posicionado o cursor é sumariamente eliminada do texto. As linhas à frente do cursor são deslocadas para cima.

**GRAPH** Caracteres para controle da impressora.

Na primeira vez que essa tecla é pressionada, faz aparecer no canto inferior da tela do vídeo a palavra GRAPH, indicando que os caracteres gráficos para controle da impressora estão ativados. Daí, é só pressionar uma das teclas numéricas de 1 a 8 para inserir no texto um símbolo gráfico para que o HOT WORD ative ou desative determinadas características especiais da sua impressora. Na segunda vez que for pressionada, volta ao modo normal de datilografia do texto.

**HOME** Centraliza o texto na linha.

O texto, na linha do cursor, é centralizado entre as margens.

- INS** Inserir um espaço em branco ou uma linha em branco.  
Caso o cursor esteja posicionado sobre uma letra, insere um espaço em branco na posição em que está o cursor. O resto da linha é deslocado para a direita. Isso, se houver espaço em branco sobrando entre a última letra da linha e a margem direita.  
Caso o cursor esteja posicionado sobre um espaço em branco, uma nova linha em branco será inserida na linha de baixo de onde está o cursor. O texto todo será deslocado para baixo. O cursor permanecerá no mesmo lugar.
- SHIFT+▲** Início do texto.  
Desloca o cursor para a primeira letra da primeira linha do texto.
- SHIFT+▼** Final do Texto.  
Desloca o cursor para a última letra da última linha do texto.
- SHIFT+◀** Desloca a linha para a esquerda.  
O trecho do texto, em uma linha, localizado à esquerda do cursor é deslocado para a esquerda. Isso se houver espaço em branco entre a margem esquerda e a primeira letra da linha. O cursor permanecerá parado no mesmo lugar.
- SHIFT+▶** Desloca a linha para a direita.  
O trecho do texto, em uma linha, localizado à direita do cursor é deslocado todo para a direita. Isso se houver espaço em branco entre a última letra da linha e a margem direita. O cursor permanecerá parado no mesmo lugar.
- ▲ Cursor para cima.  
Desloca o cursor para a linha de cima.
- ▼ Cursor para baixo.  
Desloca o cursor para a linha de baixo.
- ◀ Cursor para a esquerda.  
Desloca o cursor uma coluna para a esquerda. Se o cursor estiver na primeira coluna útil do texto, continua a deslocar por dentro da margem. Atingindo a coluna de número 1, continua na coluna de número 64 da linha de cima.

- Cursor para a direita.  
Desloca o cursor uma coluna para a direita. Se o cursor estiver na última coluna útil do texto, continua a deslocar por dentro da margem direita. Atingindo a coluna de número 64, continua na primeira coluna útil da linha seguinte.
- ◀ Cursor para a esquerda.  
Elimina uma letra ou um espaço em branco.  
Elimina a letra ou o espaço em branco sob o cursor. O trecho de texto à direita do cursor é deslocado todo para a esquerda. O cursor acompanha o deslocamento.
- RETURN** Avança o cursor para a próxima linha.  
Estando o cursor em qualquer posição de uma linha, este comando desloca-o para o início da próxima linha.  
Caso a opção de inserção automática esteja ativada, será inserida uma linha em branco.

## COMANDOS DE SERVIÇO

O Menu de Serviço apresenta várias opções de comunicação do HOT WORD com os periféricos conectados ao HOTBIT, possibilitando a transferência do texto para uma impressora, armazenamento do texto em uma fita magnética, etc.

O Menu de Serviços é acessado a partir da tela de textos com o auxílio das teclas **CTRL+STOP** e apresenta sete opções. Vamos ver para que serve cada uma delas.

### Opção P — PRINT texto

Essa opção faz com que o texto na memória do HOT WORD seja enviado para uma impressora, caso ela esteja conectada ao micro.

### Opção S — SAVE texto

Permite armazenar em uma fita cassete ou disco o texto datilografado.

### Opção L — LOAD texto

Permite transferir para a memória do HOTBIT um texto armazenado anteriormente numa fita ou disco.

### Opção M — MERGE texto

Permite transferir para a memória do HOTBIT um texto armazenado anteriormente em uma fita.

O funcionamento desta opção é semelhante ao da opção **L**, diferindo apenas num detalhe: Enquanto a opção **L** apaga o texto que porventura estiver na memória antes da transferência, esta opção mantém aquilo que já está na memória e o texto transferido é adicionado ao final do texto existente.

Caso o espaço disponível seja insuficiente, será transferido apenas um trecho até completar a capacidade da memória.  
Opção **R** — RETORNAR ao texto

Esta opção permite sair do Menu de opções e retornar ao texto na memória.

Opção **D** — DEFINIR parâmetros

Opção **T** — SAVE HOTWORD

Essa opção produz uma cópia do HOT WORD em disquete ou em fita cassete. Isso é bastante útil para salvar versões do HOT WORD preparadas para impressoras diferentes.

Para escolher qualquer opção do Menu de Serviço, basta digitar a letra correspondente a ela e a tecla **RETURN**.

# APÊNDICE I - Como usar o gravador cassete

Quando você desliga o computador tudo o que estiver armazenado na memória RAM será apagado.

Assim, se você quiser preservar os seus programas ou arquivos gerados por outros programas, deve gravá-los em fita cassete ou disco magnético antes de desligar o microcomputador. Dessa maneira será possível recarregá-los na memória posteriormente.

O sistema de armazenamento em fita cassete não é o mais perfeito possível pois apresenta muitas falhas: falta de confiabilidade, demora no tempo de gravação e leitura e acesso sequencial. Por outro lado, ele se constitui no sistema mais econômico de arquivo.

Sendo um sistema sujeito a falhas, é bastante importante que você tome alguns cuidados para minimizá-las:

1) Utilize um bom gravador.

A SHARP produz o HB-2400, que é um gravador especialmente projetado para o uso com o HOTBIT e outros microcomputadores.

2) Utilize fitas de boa qualidade, que reproduzam bem os agudos, não soltem emulsão magnética e que não sejam de longa duração.

3) Mantenha a cabeça magnética e o rolo pressor do gravador sempre limpos. Utilize álcool isopropílico para a limpeza e espere secar para introduzir a fita.

4) Desmagnetize periodicamente o cabeçote. Se você não possuir um desmagnetizador, leve o gravador em uma oficina técnica.

5) Guarde suas fitas em locais secos, limpos e afastados de fontes de interferência magnética tais como televisores, caixas acústicas, transformadores, reatores de lâmpadas fluorescentes, imãs, etc.

6) Rebobine sempre suas fitas após o uso.

7) Faça várias cópias de seus principais programas em fitas diferentes.

Para utilizar o gravador, você deve conectá-lo ao HOTBIT com o cabo que acompanha o equipamento. Esse cabo é facil-

mente identificável pois possui em uma de suas extremidades um "plug" de 8 pinos e na outra três "plugs" (P1, P1 e P2). Veja a figura 1.1.

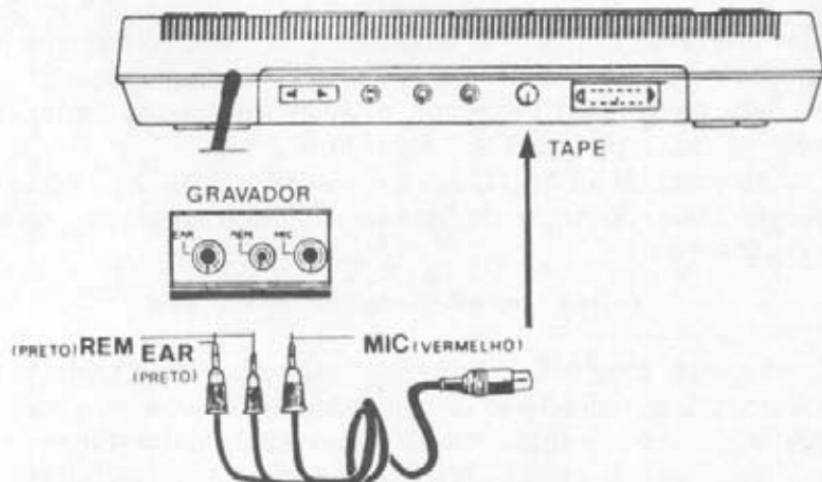
Figura 1.1



Conecte o "plug" de 8 pinos na parte posterior do micro na saída marcada com a palavra "TAPE". Note que só há uma posição em que o "plug" se encaixa.

Na outra extremidade, conecte o gravador. Os "plugs" maiores, vermelho e preto, devem ser, respectivamente ligados à entrada "MIC" e à saída "EAR" ou "MON" do gravador. Com relação ao "plug" menor (preto) ligue-o à entrada de controle remoto (REM) de seu gravador, se ele possuir uma. Veja a figura 1.2.

Figura 1.2



## COMANDOS DE ARMAZENAMENTO E LEITURA

Para gravar e carregar programas em HOT BASIC e em Linguagem de Máquina devem-se utilizar comandos específicos descritos a seguir.

### a) CSAVE e CLOAD

Para gravar em fita cassete um programa escrito em BASIC, utilize o comando CSAVE. Sua sintaxe é:

**CSAVE "nome"**

Onde, "nome" é um conjunto de no máximo 6 caracteres e deve estar sempre entre aspas ("").

O procedimento de gravação é o seguinte:

- \* Digite CSAVE "nome" mas ainda não pressione a tecla RETURN.
- \* Aclique as teclas PLAY e RECORD de seu gravador com a fita devidamente posicionada.
- \* Pressione a tecla RETURN.

A gravação estará completa quando aparecerem a palavra "Ok" e o cursor na tela.

Nesse ponto, você pode testar se a sua gravação foi bem feita utilizando o comando CLOAD?. Para isso, rebobine a fita, comande CLOAD? e acione a tecla PLAY do gravador.

Não se esqueça de ajustar o volume de seu gravador. Um nível médio, a princípio, é adequado.

Ao executar CLOAD?, deve aparecer na tela, logo no início da reprodução do trecho da fita que contém o programa gravado, a mensagem:

**Achéi: nome dado ao programa**

Se isso acontecer, tudo está correndo bem. Espere então até o final da reprodução. Deverão surgir novamente a palavra "Ok" e o cursor, indicando que a gravação está perfeita.

Se algo der errado, tente novamente com um ajuste de volume em seu gravador ligeiramente diferente do anterior.

Se após algumas tentativas, você não obtiver sucesso, é provável que a gravação não esteja boa e, nesse caso, é interessante efetuar novamente a gravação do programa armazenado na memória do micro.

Quando você quiser carregar um programa gravado com CSAVE na memória do seu HOTBIT, utilize o comando CLOAD seguindo o mesmo procedimento descrito anteriormente para o comando CLOAD?. A diferença entre os dois é que o primeiro carrega um programa na memória do micro apagando o anterior e segundo apenas verifica se o que está na memória é idêntico ao que está gravado na fita.

O comando CLOAD pode ser utilizado de duas formas:

**CLOAD** (seguido da pressão da tecla RETURN) ou  
**CLOAD"nome"** (seguido da pressão da tecla RETURN)

Na primeira forma, será lido o primeiro programa encontrado na fita.

Na segunda, será lido apenas o programa especificado pelo nome dado. Caso exista outro programa com um nome diferente gravado na fita antes do desejado, deve surgir a mensagem:

**Pulei: outro nome**

Isso acontece até que seja achado o programa especificado por "nome", quando finalmente surge a mensagem:

**Achei: nome**

Espere o fim da leitura, ou seja, aguarde a liberação do cursor e a palavra "Ok".

Novamente problemas de ajuste de volume do gravador podem ocorrer. Se você não obtiver sucesso numa primeira tentativa, não desista. Tente novamente com outros níveis de ajuste.

Se mesmo assim você não conseguir carregar seus programas, talvez seja necessário fazer um ajuste do alinhamento do cabeçote de seu gravador (azimute). Para isso, localize o furo que dá acesso ao parafuso de ajuste (ao lado da cabeça de gravação) e, desconectando todos os cabos, ponha uma fita com o programa que você quer carregar. Reproduza essa fita e gire

o parafuso de ajuste, com o auxílio de uma chave de fenda, no sentido horário ou anti-horário até obter o som mais estridente possível.

### b) SAVE e LOAD

Os comandos SAVE e LOAD tem quase a mesma finalidade que CSAVE e CLOAD. A diferença está na sintaxe e na possibilidade de introduzir-se um parâmetro que permite a execução automática do programa assim que for carregado. Além disso, a gravação é feita no formato ASCII, o que possibilita o uso do comando MERGE.

Para gravar, use:

```
SAVE "CAS: nome"
```

Para carregar, utilize:

```
LOAD "CAS:"  
ou  
LOAD "CAS:nome"
```

ou ainda,

```
LOAD "CAS:", R  
ou  
LOAD "CAS:nome", R
```

caso se queira a execução automática do programa quando ele estiver carregado.

### c) BSAVE e BLOAD

Os comandos BSAVE e BLOAD são utilizados para gravar e carregar programas em Linguagem de Máquina (uma seqüência de bytes).

Para utilizar-se BSAVE, deve-se informar ao micro o endereço de memória no qual o programa em Linguagem de Máquina está armazenado. Tanto o endereço inicial quanto o final, para que o HOT BASIC saiba exatamente qual o bloco de memória deve ser copiado na fita.

Além disso, ao usar-se esse comando, pode-se informar também o endereço em que deve começar a execução do programa, caso esse não seja igual ao endereço inicial.

Sua sintaxe é:

```
BSAVE"CAS:nome", EI, EF, EE
```

onde,

EI = endereço inicial

EF = endereço final

EE = endereço de início da execução

Para carregar um programa em Linguagem de Máquina, use o comando BLOAD cuja sintaxe é:

**BLOAD"CAS:" ou BLOAD"CAS:nome"**

Se você desejar a execução automática do programa após o carregamento, basta acrescentar a letra R, ou seja:

**BLOAD"CAS:",R**

**ou**

**BLOAD"CAS:nome",R**

## APÊNDICE II - Caracteres do HOTBIT

O HOTBIT possui 256 caracteres numerados de 0 a 255 (em decimal) ou de 00H a FFH (em hexadecimal). Na tabela abaixo você pode observá-los relacionados a seus códigos numéricos em hexadecimal. Por exemplo, a letra "A" está no cruzamento da linha 4 com a coluna 1. Isso significa que seu código numérico em hexadecimal é 41. Se você comandar:

```
PRINT CHR$(&H41)
```

O caractere "A" será impresso na tela.

Isso funciona com todos os caracteres de código maior que 32 (em decimal) ou 20H (em hexadecimal). Os caracteres de códigos menores são usados para controlar funções do vídeo, da impressora, e de outros periféricos. Para imprimí-los é necessário usar, antes, o caracter de código 1. Por exemplo, se quisermos imprimir na tela o "coraçãozinho", cujo código em hexadecimal é 03H, devemos comandar:

```
PRINT CHR$(1) + CHR$(64+&H03)
```

Figura II.1 — Tabela de caracteres do HOTBIT.

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	©	®	♥	♦	±	+	-	□	○	■	♂	♀	♪	♫	*	
1	+	-	—	—	—	—	—	—	—	—	—	—	—	X	/	—
2	!	"	#	\$	%	&	'	(	)	*	+	,	-	.	/	
3	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4	À	Á	Ç	É	Ó	Í	Ô	Õ	À	Á	Ç	É	Ó	Í	Ô	Õ
5	Þ	Ŕ	Ŗ	Ŗ	Ŗ	Ŗ	Ŗ	Ŗ	Ŗ	Ŗ	Ŗ	Ŗ	Ŗ	Ŗ	Ŗ	Ŗ
6	‘	’	a	b	c	d	e	f	g	h	i	j	k	l	m	n
7	ó	ô	õ	õ	õ	õ	õ	õ	õ	õ	õ	õ	õ	õ	õ	õ
8	ç	é	à	â	ê	î	ô	û	ë	ë	ë	ë	ë	ë	ë	ë
9	é	æ	ø	ö	ö	ö	ö	ö	ü	ü	ö	ö	ö	ö	ö	ö
A	á	í	ó	ú	ñ	ä	ö	ü	ç	ç	í	ó	ú	ñ	ä	ö
B	â	ã	õ	õ	õ	õ	õ	õ	õ	õ	õ	õ	õ	õ	õ	õ
C	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
D	◀	▶	◀	▶	◀	▶	◀	▶	◀	▶	◀	▶	◀	▶	◀	▶
E	α	β	Γ	Π	Σ	σ	λ	τ	θ	Ω	δ	ω	φ	ε	η	
F	±	≥	≤	∫	J	÷	≈	°	•	-	Γ	π	2	1	—	

Os caracteres de controle estão relacionados na tabela da figura II.2, junto com seus códigos (decimal e em hexadecimal) e com suas respectivas funções no vídeo do HOTBIT.

Figura II.2 — Caracteres de controle e funções no vídeo.

TECLADO	DEC	HEX	FUNÇÃO
CTRL+A	01	&H01	Determina caracter gráfico.
CTRL+B	02	&H02	Desvia o cursor para o início da palavra anterior.
CTRL+C	03	&H03	Encerra a condição entrada.
	04	&H04	
CTRL+E	05	&H05	Cancela caracteres do cursor até o fim da linha.
CTRL+F	06	&H06	Desvia o cursor para o início da palavra seguinte.
CTRL+G	07	&H07	Aciona BEEP.
CTRL+H	08	&H08	Apaga a letra anterior ao cursor (BS).
CTRL+I	09	&H09	Move o cursor para a próxima tabulação (TAB).
CTRL+J	10	&H0A	Muda de linha (LINE FEED).
CTRL+K	11	&H0B	Leva o cursor para a posição (1,1) (HOME).
CTRL+L	12	&H0C	Limpa a tela e leva o cursor para a posição (1,1) (CLS).
CTRL+M	13	&H0D	Retorno do carro (RETURN).
CTRL+N	14	&H0E	Move o cursor para o fim da linha.
	15	&H0F	
	16	&H10	
	17	&H11	
CTRL+R	18	&H12	Ativa e desativa o modo de inserção (INS).
	19	&H13	
	20	&H14	
CTRL+U	21	&H15	Apaga toda a linha em que está o cursor.
	22	&H16	
	23	&H17	

CTRL+X	24	&H18	Equivale à tecla SELECT.
	25	&H19	
	26	&H1A	
CTRL+[	27	&H1B	Equivale à tecla ESC.
CTRL+\	28	&H1C	Move o cursor para a direita.
CTRL+]	29	&H1D	Move o cursor para a esquerda.
CTRL+^	30	&H1E	Move o cursor para cima.
CTRL+_	31	&H1F	Move o cursor para baixo.
	127	&H7F	Apaga a letra que está sob o cursor (DEL).

## FORMATAÇÃO DE TELA COM ESC

O HOT-BASIC possui comandos de formatação de tela e configuração de cursor característicos do padrão MSX, portanto incompatíveis com outros BASIC's.

Para que programas escritos em HOT-BASIC que possuam comandos de formatação de tela possam ser compatibilizados com compiladores diferentes do padrão MSX, existe a alternativa de implementá-los com seqüências de caracteres precedidos pelo "ESC" (CHR\$(27)).

Por exemplo, no lugar de posicionar o cursor pelo comando LOCATE (característico do MSX), podemos comandar:

```
PRINT CHR$(27); "Y";CHR$(núm.da linha + 32);
CHR$(núm.da coluna + 32);
```

O comando acima pode ser utilizado, por exemplo, no compilador MBASIC.

A seguir, são relacionadas as seqüências de caracteres e seus efeitos.

### 1 — Movimentação de Cursor

CHR\$(27); "A";	Move o cursor para cima
CHR\$(27); "B";	Move o cursor para baixo
CHR\$(27); "C";	Move o cursor para direita
CHR\$(27); "D"	Move o cursor para esquerda
CHR\$(27); "H";	Leva o cursor para as coordenadas (1,1)
CHR\$(27); "Y";	
CHR\$(L + 32);	
CHR\$(C + 32);	

## **2 — Edição**

CHR\$(27);"j";	Equivale ao CLS (limpa a tela)
CHR\$(27);"E";	Equivale ao CLS (limpa a tela)
CHR\$(27);"K";	Apaga do cursor ao fim da linha
CHR\$(27);"J";	Apaga do cursor ao fim da tela
CHR\$(27);"I";	Apaga a linha sob o cursor
CHR\$(27);"L";	Insere uma linha
CHR\$(27);"M";	Apaga uma linha

## **3 — Configuração do cursor**

CHR\$(27);"x";"4";	Muda para cursor cheio (8x8)
CHR\$(27);"x";"5";	Faz o cursor aparecer
CHR\$(27);"y";"4";	Muda para meio cursor (4x8)
CHR\$(27);"y";"5";	Faz o cursor desaparecer

## APÊNDICE III - Mensagens de Erro

Código	Mensagem	Descrição
54	Arquivo aberto	Tenta-se abrir, através de uma instrução "OPEN" um arquivo que já tinha sido aberto, ou então dá-se um KILL para um arquivo aberto.
53	Arquivo não existe	Um comando LOAD, KILL ou OPEN faz referência a um arquivo não existente no disco.
58	Arquivo seqüencial	Uma instrução própria para o acesso aleatório é dada para um arquivo seqüencial.
50	Campo maior	Uma instrução FIELD tentou posicionar maior número de bytes dos que foram especificados para o comprimento "record" de um arquivo aleatório (255 bytes). Ou então, o fim de um buffer "FIELD" foi encontrado enquanto estava-se fazendo I/O seqüencial (PRINT#INPUT#) a um arquivo aleatório.
57	Comando direto no Arquivo	Foi encontrada uma instrução direta enquanto estava sendo carregado (LOAD) um arquivo em forma ASCII. O carregamento do arquivo é terminado.
10	DIM redefinido	Duas instruções DIM foram dadas para a mesma matriz, ou uma instrução DIM foi dada para uma matriz depois que a dimensão "default" de 10 foi fixada para essa mesma matriz.
12	Direto ilegal	Uma instrução que é ilegal no modo direto foi introduzida como comando no modo direto.
11	Divisão por zero	Foi encontrada uma divisão por zero em uma expressão ou então uma situação na qual o zero é elevado a uma potência negativa.
23	Erro indefinido	Não existe mensagem de erro disponível para a condição de erro existente. Isto é causado normalmente por um ERROR, com um código de erro indefinido.

Código	Mensagem	Descrição
51	Erro interno	Erro provocado por alguma situação não prevista pelo interpretador BASIC.
19	Erro periférico	Aconteceu um erro I/O em uma operação de disco, cassete, impressora ou TRC. Este é um erro fatal, isto é, o BASIC não consegue se recuperar deste tipo de erro.
2	Erro sintaxe	Foi encontrada uma linha contendo uma sequência incorreta de caracteres (por exemplo: falta fechar parênteses, comando com erro de sintaxe, sinais de pontuação errados, etc.).
20	Erro verificação	O programa na memória é diferente do programa salvo na fita cassete.
14	Falta área STRING	As variáveis string fizeram com que o BASIC supere o espaço de memória disponível. O BASIC posiciona os strings dinamicamente conforme as necessidades, até ficar sem memória disponível. Pode-se usar a instrução CLEAR para criar maior espaço livre.
7	Falta memória	O programa é muito longo, tem excesso de arquivos, tem excesso de laços FOR ou GOSUB, tem excesso de variáveis ou expressões muito complexas.
59	Falta OPEN	O arquivo especificado em um PRINT#INPUT# etc., não foi aberto através do OPEN correspondente.
24	Falta operando	Foi encontrada uma expressão que contém um operador sem o operando correspondente a seguir.
55	Fim do arquivo	Foi executada uma instrução INPUT após todos os dados já terem sido introduzidos, ou para um arquivo nulo (vazio). Para evitar este erro, usa-se a função EOF a fim de detectar o fim do arquivo.

Código	Mensagem	Descrição
5	Função ilegal	Um parâmetro que está fora da faixa foi passado para uma função matemática ou para uma função string. Esse erro também pode acontecer como resultado de:
		<ol style="list-style-type: none"> <li>1. Um índice negativo ou extremamente grande.</li> <li>2. Um argumento negativo ou zero usado para LOG.</li> <li>3. Um argumento negativo usado para a SQR.</li> <li>4. Um argumento impróprio usado em MID\$, LEFT\$, RIGH\$, INP, OUT, PEEK, POKE, TAB, SPC, STRINGS\$, SPACE\$, INSTR\$ ou ON...GOTO</li> </ol>
18	Função indefinida	Foi chamada uma função definida pelo usuário (FN) sem antes tê-la definido através de uma instrução DEF FN.
9	Índice fora do Limite	Um elemento da matriz foi definido com um índice que está fora das dimensões pré-fixadas para essa matriz ou então com um número errado de índices.
25	Linha Longa	Foi introduzida uma linha com excessivo número de caracteres.
17	Não continuo!	Tentou-se continuar um programa que: <ol style="list-style-type: none"> <li>1. Tinha sido detido devido a um erro.</li> <li>2. Foi modificado durante um "BREAK" na execução, ou</li> <li>3. Não existe.</li> </ol>
1	NEXT sem FOR	Em uma instrução "NEXT" uma variável não se corresponde com nenhuma outra previamente executada, ou então uma instrução "NEXT" que não é precedida pela FOR correspondente.
56	Nome do arquivo	Foi utilizada uma forma ilegal na definição do nome do arquivo usado em LOAD, SAVE, KILL, NAME, etc.

Código	Mensagem	Descrição
52	N.º do arquivo	Algum comando ou instrução faz referência a um arquivo para o qual não foi dado previamente o OPEN ou então o N.º está fora da faixa fixada para o N.º de arquivos, através do MAXFILES.
8	N.º linha inexistente	Faz-se referência a uma linha inexistente em uma instrução GOTO, GOSUB, IF... THEN..., ELSE.
6	Overflow	O resultado de um cálculo é grande demais para ser representado no formato numérico do BASIC .
21	RESUME	O BASIC entrou numa rotina de tratamento de erros que não contém a instrução RESUME.
22	RESUME sem ERROR	Foi encontrada uma instrução RESUME antes de ter entrado numa rotina de tratamento de erros.
3	RETURN sem GOSUB	Foi encontrada uma instrução RETURN sem a sua correspondente GOSUB prévia.
4	Sem DATA	Foi executada uma instrução READ quando já não existe instrução DATA com dados ainda não lidos durante o programa.
16	STRING complexa	Foi criada uma expressão "string" excessivamente longa ou complexa. A expressão deverá ser partida em expressões menores.
15	STRING longa	Tentou-se criar uma "string" com mais de 255 caracteres.
13	Tipo desigual	Foi atribuído um valor numérico a uma variável string ou vice-versa, ou então, uma função que espera um argumento numérico recebe um argumento string ou vice-versa.
26 a 49		Esses códigos não tem definições. Foram reservados para futuras expansões do BASIC.

## APÊNDICE IV - Funções Matemáticas

As funções matemáticas derivadas que não foram incluídas no HOT-BASIC, podem ser definidas da seguinte forma:

**DEFFN nome da função(X) = função escolhida na figura IV.1**

O nome escolhido para a função pode ter um número quaisquer de caracteres, mas apenas os dois primeiros são considerados pelo HOT-BASIC.

Para utilizar a função definida, use:

**FN nome da função(parâmetro)**

Por exemplo, suponha que se queira definir uma função que calcule a raiz quadrada de um número elevado ao cubo.

Se "batizarmos" essa função de RQC, então podemos defini-la e utilizá-la conforme o programa a seguir:

```
10 DEFFN RQC(X)=SQR(X^3)
20 INPUT "Entre um numero ";N
30 PRINT FN RQC(N)
```

Observação: as funções apresentadas na figura IV.1 que contêm a constante PI (3.141592...), devem ser utilizadas após a definição dessa constante. Para isso, inclua em seu programa a linha a seguir:

```
XXXXX PI = 4*ATN(1)
```

Onde, XXXXX é o número da linha do programa que deve anteceder qualquer outra que utilize a constante PI.

Figura IV.1 — Funções Matemáticas

FUNÇÕES DERIVADAS	DEFINIÇÃO PARA O BASIC
BECANTE	1/COS(X)
COSSECANTE	1/SIN(X)
COTANGENTE	1/TAN(X)
ARCO SENO	ATN(X/SQR(-X*X+1))
ARCO COSENTO	-ATN(X/SQR(X*X-1))+PI/2
ARCO SECANTE	ATN(X/SQR(X*X-1))+SGN(SGN(X)-1)*PI/2
ARCO COSSECANTE	ATN(X/SQR(X*X-1))+(SGN(X)-1)*PI/2
ARCO COTANGENTE	ATN(X)+PI/2
SENO HÍPERBÓLICO	(EXP(X)-EXP(-X))/2
COSSENO HÍPERBÓLICO	(EXP(X)+EXP(-X))/2
TANGENTE HÍPERBÓLICA	EXP(-X)/EXP(X)+EXP(-X)*2+1
SECANTE HÍPERBÓLICA	2/(EXP(X)+EXP(-X))
COSSECANTE HÍPERBÓLICA	2/(EXP(X)-EXP(-X))
COTANGENTE HÍPERBÓLICA	EXP(-X)/EXP(X)-EXP(-X)*2+1
ARCO SENO HÍPERBÓLICO	LOG(X+SQR(X*X+1))
ARCO COSENTO HÍPERBÓLICO	LOG(X+SQR(X*X-1))
ARCO TANGENTE HÍPERBÓLICA	LOG((1+X)/(1-X))/2
ARCO SECANTE HÍPERBÓLICA	LOG((SQR(-X*X+1)+1)/X)
ARCO COSSECANTE HÍPERBÓLICA	LOG((SGN(X)*SQR(X*X+1)+1)/X)
ARCO COTANGENTE HÍPERBÓLICA	LOG((X+1)/(X-1))/2

## APÊNDICE V - Uso da Impressora

O padrão MSX, no Brasil, prevê 256 caracteres diferentes. Os números, os principais sinais de pontuação, as letras maiúsculas e minúsculas são impressos por praticamente todas as impressoras disponíveis no mercado (inclusive as importadas).

O problema começa a surgir com o "ç" e as vogais acentuadas. Algumas impressoras não têm condições de imprimí-los, outras os têm, mas identificados por códigos diferentes do MSX.

Se você colocar em seus programas apenas caracteres não acentuados, não deve ter maiores preocupações com a impressora e pode dispensar a leitura deste apêndice.

Se quiser sofisticar seus programas ou textos produzidos por programas tais como o HOT WORD, HOT PLAN e HOT DATA, então convém ler com cuidado o que vem a seguir.

A tabela de caracteres residente no seu HOTBIT é a descrita no Apêndice II.

Nas impressoras brasileiras existem, principalmente, três tipos de tabelas de caracteres:

MSX

ABNT

ABICOMP

Os caracteres que vão do código 32 ao 126 (sinais de pontuação, aritméticos, financeiros, algarismos e letras maiúsculas e minúsculas do alfabeto inglês) são comuns às 3 tabelas.

Não usando outros caracteres, basta conectar sua impressora (paralela CENTRONICS) ao seu HOTBIT sem maiores problemas.

Vamos agora descrever os procedimentos que você deve adotar para obter os demais caracteres em função do tipo de impressora:

### a) IMPRESSORA MSX

Antes de mandar dados para a impressora comande:

**POKE &HF417, 1**

Esse mesmo comando deve ser dado quando a impressora é colocada no modo gráfico para fazer cópias, por exemplo, da tela de alta resolução.

Programas que fazem esse tipo de cópias podem ser encontrados nos livros: "APROFUNDANDO-SE NO MSX" e "PROGRAMAÇÃO AVANÇADA EM MSX", citados na bibliografia do capítulo 2.

### b) IMPRESSORA ABICOMP

Seu HOTBIT já está configurado para enviar dados para a impressora segundo o padrão ABICOMP.

Se por algum motivo ele foi tirado, via software, dessa configuração, para restabelecê-la, basta comandar:

```
POKE &HF417,0
```

### c) IMPRESSORA ABNT

O código BRASCI foi estabelecido pela ABNT (Associação Brasileira de Normas Técnicas) e passou a ser utilizado como padrão das impressoras mais recentes.

O seu HOTBIT, graças à extrema elasticidade de sua arquitetura, pode ser configurado para enviar caracteres segundo esse padrão. Para isso ele deve ser previamente carregado com um "programa-filtro". Para maiores detalhes, aconselhamos a leitura do livro "PROGRAMAÇÃO AVANÇADA EM MSX" citado no capítulo 2.

## **AGRADECIMENTOS**

Os autores agradecem ao Sr. José Mário Fonseca de Andrade, Sra. Elisa Cazartelli e Sr. Otto Longo Sernatinger pela valiosa contribuição na elaboração deste livro. Para o Prof. Wilson José Tucci registraram um agradecimento especial pelo auxílio prestado na elaboração do capítulo referente ao LOGO.

## **LITERATURA PARA MSX**

Para obter mais informações sobre livros para seu HOTBIT, envie seu nome e endereço para

**EDITORIA ALEPH**  
CP 20.707  
01498 S. Paulo SP

**Este livro foi impresso com  
fotolitos fornecidos pela Editora  
Gráfica Palas Athena  
Associação "Palas Athena" do Brasil  
Rua José Bento, 384  
Fone: 279-6288 — CEP 01523  
Cambuci — São Paulo**

# COLEÇÃO MSX

## COMO USAR O SEU HOT BIT

**COMO USAR SEU HOTBIT** apresenta ao leitor, numa linguagem clara e acessível, uma visão geral dos fantásticos recursos do microcomputador MSX da SHARP. Saindo dos padrões tradicionais, os autores, logo após explicarem a instalação e o manuseio da máquina, iniciam o leitor no universo da programação através da linguagem LOGO, desenvolvida especialmente para aplicações didáticas e pedagógicas.

Em seguida, explicam como usar a linguagem residente do micro, o HOT BASIC, uma versão das mais completas e poderosas do BASIC. Os capítulos finais do livro mostram como usar os três aplicativos mais utilizados em microcomputadores: o processador de textos (HOT WORD), o banco de dados (HOT DATA) e a planilha de cálculos (HOT PLAN). Complementando a obra, existem vários apêndices com dados e informações gerais sobre o HOTBIT. Um livro indispensável para quem tem ou quer ter um micro MSX!