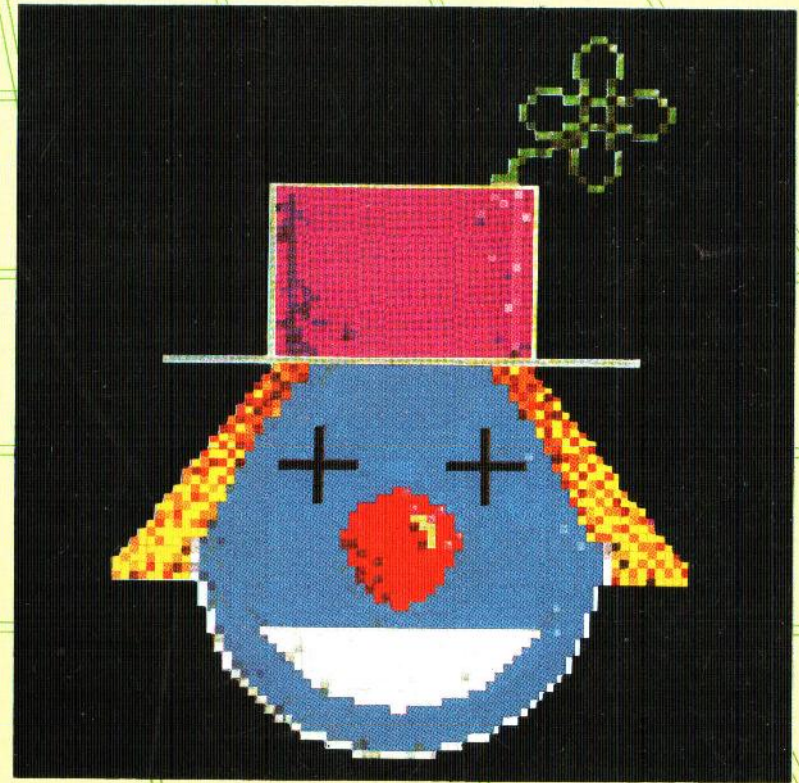


COLEÇÃO MSX



HOTLOGO

MSX

# HOTLOGO

PRIMEIROS PASSOS

Gráfica Palas Athena  
Associação "Palas Athena" do Brasil  
Rua José Bento, 384  
Fone: 279-6288 — CEP 01523  
Cambuci — São Paulo

**SHARP**



# HOTLOGO

## PRIMEIROS PASSOS

ANA LUISA LACERDA

MARCOS LEPÍSCOPO

MÔNICA HOEHNE MENDES

NORMA DIAS DE ARAUJO COSTA GODOY

**COLEÇÃO MSX**



1ª EDIÇÃO

© - 1986 EDITORA ALEPH

A reprodução de quaisquer trechos desta obra para fins exclusivamente didáticos é permitida desde que autorizada por escrito pela Editora.

Direção Editorial: Pierluigi Piazzzi  
Coord. Pedagógica: Betty F. Piazzzi  
Editoração: Renato da Silva Oliveira  
Arte: Ana Lúcia Antico  
Ilustrações: Durvaly Odilon Nicoletti  
Capa: Alvaro Guillermo  
Produção: Rosa Kogan Fromer



ALEPH Publicações e Ass. Ped. Ltda.  
Av. Brig. Faria Lima, 1451 - Conj. 31  
01451 - São Paulo - SP  
Tel.: (011) 813-4555 ou 212-4917  
Caixa Postal: 20707 - CEP:01498

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

L134h	Lacerda, Ana Luísa. HOTLOGO : primeiros passos / Ana Luísa Lacerda ... (et al.). -- 1. ed. -- São Paulo : Aleph, 1986. (Coleção MSX)
	1. HOTLOGO (Computador) I. Título. II. Série.
86-1270	CDD-001.64

Índices para catálogo sistemático:

1. HOTLOGO : Computadores : Processamento de dados 001.64

## SUMÁRIO

Nota aos pais e professores .....	7
Introdução .....	9
Primeiras Noções .....	15
Conhecendo a tela de trabalho .....	19
Traçando formas simples .....	27
Usando cores .....	35
Construindo polígonos .....	41
Realizando projetos .....	47
Criando novos comandos .....	53
Gerando arquivos .....	61
Como fazer contas com o HOTLOGO .....	65
Aprendendo variáveis .....	67
Decompondo um projeto em partes .....	73
Elaborando procedimentos recursivos .....	77
Multiplicando figuras .....	81

Trabalhando com SPRITES .....	89
Som e animação .....	97
Ultimas considerações .....	101
Comandos comentados .....	103
Lista de primitivas .....	115
Apêndice .....	131

## NOTA AOS PAIS E PROFESSORES

Há alguns anos toda uma geração de pais e professores viveu o drama da introdução da famosa "matemática moderna" nas escolas. Houve até o caso de pais que tomaram aulas particulares sobre teoria dos conjuntos para poder ajudar os filhos na lição de casa!

Hoje vivemos uma fase de transição muito mais drástica porém muito mais divertida. A "febre" da computação tomou conta dos lares e das escolas.

Como toda mudança esta febre traz consigo muitas vantagens e muitos problemas. Entre os problemas podemos citar a introdução apressada e imediatista do microcomputador na escola, mais para atender uma necessidade artificial de "modernização" do que para desenvolver uma válida atividade pedagógica.

Como consequência vimos a proliferação de cursos ineficientes, dados por professores improvisados, recrutados entre técnicos, programadores ou até "curiosos" sem o mínimo preparo ou conhecimento didático-pedagógico. Quando estes cursos são dados a crianças, então, os estragos são bem maiores e às vezes irreparáveis.

Um assunto gostoso, e intelectualmente muito benéfico, se mal introduzido, pode produzir um processo de rejeição que fecha as portas de todo um universo de maravilhosas

descobertas.

Por isso tomamos o cuidado de encomendar este livro a uma equipe de pedagogos conscientes da importância de uma iniciação estimulante neste mundo estranho, novo e maravilhoso da computação.

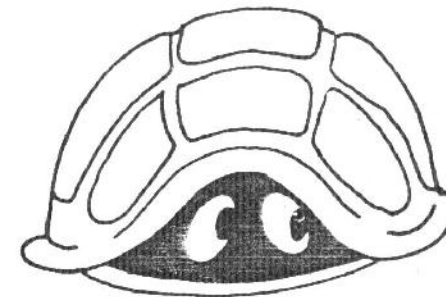
A linguagem LOGO, da qual o HOT-LOGO representa uma das mais felizes versões, foi desenvolvida com base em critérios científicos que levam em consideração todas as etapas de desenvolvimento intelectual da criança. Representa, portanto, uma das formas mais suaves e naturais para se iniciar este processo de aprendizado.

A grande vantagem deste aprendizado não consiste, como muitos imaginam, na simples incorporação de uma nova técnica. Ela reside, isto sim, num real desenvolvimento intelectual da criança, aumentando sua capacidade de raciocínio e estruturação mental.

A tarefa de desenvolver o raciocínio da criança foi atribuída, até agora, à cadeira de matemática. Falando francamente, não podemos dizer que ela tenha sido cumprida a contento. O surgimento do LOGO nos dá uma segunda chance e de maneira muito mais abrangente.

Finalmente a escola tem condições de desempenhar sua verdadeira (e talvez única) finalidade: ENSINAR A PENSAR!

## INTRODUÇÃO



*Computador!... que caixa é esta tão cheia de mistérios que, quando usada por pessoas mais velhas, parece saber todas as respostas?*

*Entretanto, quando tento, tudo é complicado e sem sentido... Como fazer para falar com ele? Que linguagem usar para conseguir uma forma de comunicar-me com esta máquina?*

Sem dúvida, você já se deve ter feito todas estas perguntas.

Na verdade, não existem soluções imediatas, como para todas as coisas da nossa vida. O importante é descobrir a maneira de nos comunicar com o computador.

Existem muitos meios de comunicação. O mais elementar deles é a fala e, com ela, deixamos transparecer emoções como alegria, tristeza, calma, raiva, etc... Bem, então você aprende uma língua para comunicar-se, que é o português. Mas, o que acontece se você ganha de presente uma viagem para outro país? Será que ao chegar lá as pessoas vão lhe entender?

Não, evidente! Você vai precisar conhecer algumas palavras no idioma daquele país para poder estabelecer um diálogo com qualquer pessoa que lá encontrar.

Em computação acontece o mesmo. Precisamos aprender a linguagem do computador.

Compare o computador ao mundo. Há no mundo vários povos que falam diferentes linguagens. Como ele, o computador também possui diferentes linguagens, sendo que a maioria delas é muito difícil para ser usada por crianças.

Foi pensando nas crianças que, há algum tempo, um homem decidiu criar uma linguagem que lhes permitisse trabalhar com o computador.

Isto não quer dizer que seja simples demais para os adultos e posso garantir que eles também vão gostar muito de trabalhar com ela.

Vejamos então quem é este homem e que linguagem é esta.

Seu nome é Seymour Papert e a linguagem por ele criada é o LOGO.

Ao usar esta linguagem você entrará num mundo mágico, cheio de novidades e até aventuras.

O mais interessante é que você não estará sozinho nesses momentos. Você terá uma companheira que obedecerá sempre às suas ordens, de maneira rápida e eficiente.

Você prestou atenção ao que acabou de ler?

Sua futura amiga somente poderá executar aquilo que você lhe pedir pois depende totalmente de suas instruções.

Gostaria de conhecê-la agora?



Isso mesmo! Sua companheira é uma tartaruga!

Uma tartaruga que possui um lápis embaixo de seu casco e que deixa um rastro por onde passa.

Através dela você poderá realizar seu grande desejo:

comunicar-se com o computador.

Mas, atenção!

A tartaruga inicialmente conhece apenas algumas palavras que, de uma forma mais técnica, são denominadas COMANDOS PRIMITIVOS.

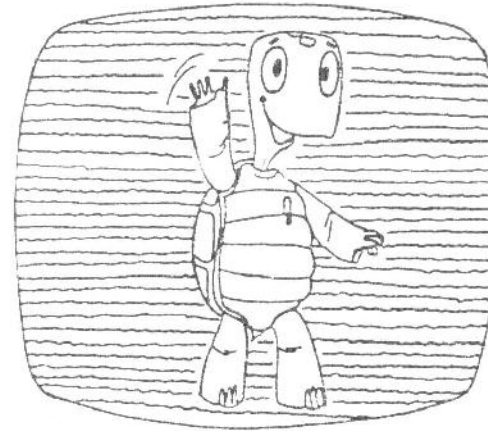
Da mesma forma que o ser humano, a partir do momento em que nasce, vai crescendo e aprendendo novas expressões, sua companheira poderá aumentar o seu vocabulário, através de novas palavras que chamamos de PROCEDIMENTOS.

O mais divertido de tudo isso é que a você cabe a importante missão de fazer a tartaruga conhecer este maior número de palavras. Portanto, em pouco tempo você vai ser o seu professor.

Está parecendo muito complicado?



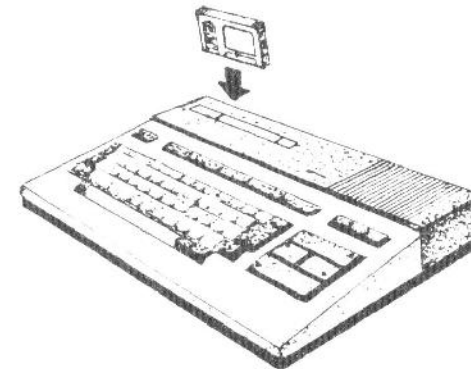
Talvez a primeira vista, seja esta a impressão, porém, se você acompanhar cada etapa proposta por este livro, obedecendo a sequência de seus capítulos, verá que de maneira gostosa e interessante poderá obter resultados incríveis, como por exemplo, criar um desenho animado.



## PRIMEIRAS NOÇÕES

Para começar a usar o HOTLOGO em seu micro, siga corretamente estes passos:

- com o computador desligado, coloque o cartucho no local adequado;



- ligue o computador e a TV; espere alguns segundos e em sua tela surgirá a mensagem escrita a seguir:



Repare neste quadradinho branco que está piscando. É o cursor. Ele marca o local onde aparecerá qualquer letra ou sinal correspondente a uma tecla por você digitada.

A palavra "digitada" é nova para você?

Em computação, podemos dizer que digitar significa teclar, ou melhor, apertar uma tecla.

É digitando, no teclado, as palavras conhecidas pela tartaruga, que podemos estabelecer a nossa comunicação com o HOTLOGO.

Antes de aprender que palavras são essas, precisamos conhecer melhor o teclado.

A primeira vista ele é semelhante ao teclado de uma máquina de escrever, embora possua pequenas diferenças.

Vamos falar agora sobre algumas teclas e o que fazem. Outras, serão vistas em capítulos futuros.

Sua expectativa para mexer no computador deve estar grande e você poderia começar a usá-lo digitando seu nome.

Feito isso, aperte RETURN. Sempre que você usar esta tecla, estará pedindo ao HOTLOGO que leia as instruções digitadas, executando-as quando possível. Caso contrário, ele lhe devolverá uma mensagem contendo a explicação do porque não ter obedecido as suas ordens.

Suponha que você digitou uma palavra, trocando suas letras, como por exemplo, PEDAR, ao invés de PEDRA. Para corrigir este erro, use esta tecla ◀◀, que fica logo acima de RETURN.

O que ela faz?

Apaga sempre o que estiver à esquerda do cursor.

Como fazer para digitar um dos sinais encontrados acima dos números? É preciso pressionar primeiro a tecla SHIFT, mantê-la assim e, junto, a tecla com o sinal desejado. Com a prática você acabará apertando as duas teclas simultaneamente.

O HOTLOGO só entenderá os comandos escritos em letra minúscula. Por isto, você deve tomar cuidado com CAPS, pois ela altera o padrão da letra para minúscula ou maiúscula.

As teclas ◀ e ▶, como a própria figura indica, deslocam o cursor para a esquerda ou para a direita, permitindo fazer correções.

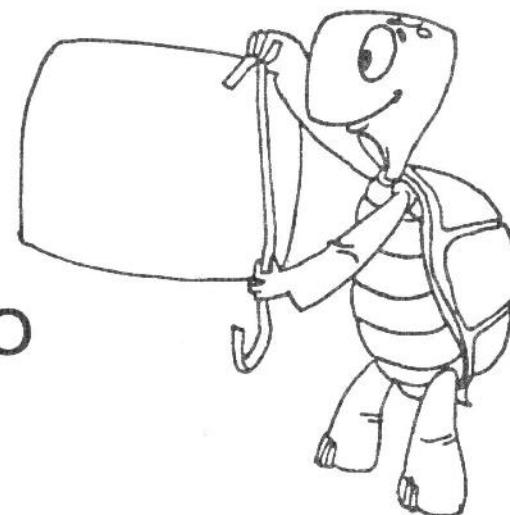
Já as teclas ▲, ▼ e ESC são usadas quando estamos ensinando ou corrigindo procedimentos, o que veremos mais adiante.

Restam apenas duas teclas: CTRL e DEL. A primeira, quando acionada com outra tecla, altera a função das mesmas, e a segunda apaga o caracter que estiver embaixo do cursor.

As demais teclas especiais existentes no seu microcomputador não são utilizadas pelo HOTLOGO.

Embora estas informações possam parecer cansativas, são necessárias.

Descanse e prepare-se para entrar, em seguida, no Mundo Mágico do LOGO.



## CONHECENDO A TELA DE TRABALHO

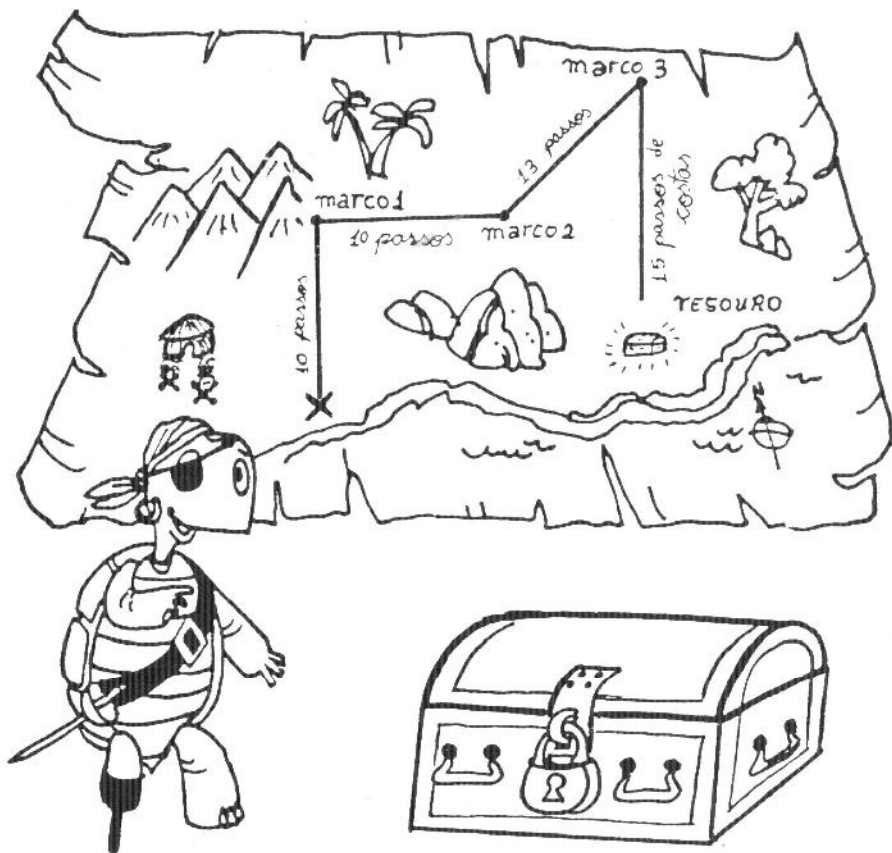
No capítulo anterior vimos como usar o teclado do micro, embora não tenhamos, ainda, aprendido a fazer a tartaruga "andar".

Você lembra que já falamos sobre a tartaruga conhecer apenas os **COMANDOS PRIMITIVOS**?

Muito bem. Vejamos, então, os quatro comandos que permitem dar movimento à tartaruga.

Ah! Se você pensa que vamos simplesmente dizer quais são eles, enganou-se! Você terá que tentar descobri-los, brincando conosco.

A proposta é chegar a um tesouro perdido seguindo o mapa desenhado no início da próxima página:



Bom! Para conseguir fazer isto no computador é necessário que primeiro você realize os movimentos sugeridos no mapa num espaço qualquer, como por exemplo, no tapete da sala, no play-ground, etc....

Agora pense no que você está fazendo: ao deslocar-se 10 passos, a partir do local assinalado com um X, até o marco 1, em que direção andou?

-----

Para chegar ao marco 2 é necessário que você faça dois movimentos. Diga quais são:

-----

Do marco 2 ao marco 3 são necessários mais dois movimentos, mas apenas um é diferente dos anteriores.

Que movimento é este?

-----

Você já percorreu mais da metade do caminho. Agente firme! Falta pouco! Agora gire um pouco para a esquerda ficando na posição correta para realizar a próxima etapa que tem uma novidade:

andar de marcha-à-ré.

Como é que você pode chamar, este andar, de outra forma?

-----

Pronto! Depois de tudo isto você já deve ter descoberto as quatro palavras-chave, mas vamos conferir o caminho percorrido:

Primeiro você andou para frente, em seguida virou-se para direita e novamente caminhou para frente. Depois você se moveu um pouco para esquerda, seguiu em frente e virou outra vez para esquerda. Sua próxima instrução dizia para você dar 15 passos para trás.

Reparou que os comandos foram escritos de forma diferente da que você está acostumado, isto é, juntamos duas palavras numa só? Este é um aspecto importante da linguagem. Todo comando, ainda que formado por duas palavras ou mais, só será entendido pelo LOGO se escrito sem espaço entre elas.

Ufa! Que comandos compridos! Eles podem ser simplificados desta forma:

COMANDO	FORMA ABREVIADA
para frente	pf
para trás	pt
para direita	pd
para esquerda	pe

Faça finalmente sua amiga caminhar pela tela agora que já conhece os comandos que lhe dão movimento.

Só mais uma coisa...

Se você foi mais rápido do que eu e já tentou usar um dos comandos acima, teve como resposta esta mensagem:

Não há entradas suficientes para ...

Voltemos ao mapa. Para ir de um marco ao outro era necessário dar os passos nele indicados bem como girar numa determinada direção. Nossa amiga somente poderá executar qualquer um dos comandos acima quando estes vierem acompanhados de um número que determine a distância que ela deve percorrer ou o quanto ela deve girar.

Estes números funcionam como a entrada necessária a realização destes comandos e respondem a mensagem obtida na tela.

Agora sim, não falta mais nada!

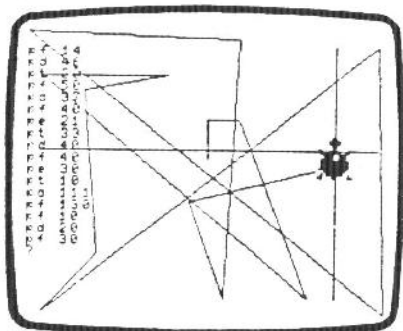
Tudo o que lhe cabe fazer neste momento é brincar bastante com estes quatro comandos, fazendo com que a tela seja bem explorada por sua auxiliar.

Dê uma paradinha e responda: há alguma diferença entre os comandos pf/pt e pd/pe? Se sua resposta foi positiva, tente explicar que diferença é esta.

-----  
-----

Confira comigo mais uma vez: sim, há uma diferença entre os comandos pf/pt e pd/pe. Os primeiros fazem com que a tartaruga ande quantos passos lhe pedirmos e os segundos fazem com que ela gire na direção solicitada.

Posso supor que depois de muito brincar sua tela esteja cheia de traços e comandos, mais ou menos parecida com esta:



Concordo que fica tudo muito confuso e sem graça. Por que isto acontece? Porque é como se estivéssemos trabalhando com duas telas, sendo uma em cima da outra. Com o comando att fazemos as letras sumirem, apagando a tela de texto e ficando apenas com a visão da tela gráfica.

Para conseguir o inverso, ou seja, apagar a tela gráfica mantendo o texto, use ad que significa apague desenho.

O que acha de limparmos tudo? Neste caso você poderá apagar as duas telas, simultaneamente, se recorrer ao comando tat.



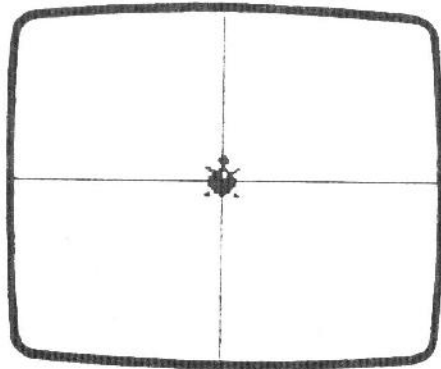
Verifique, na lista de comandos encontrada no final do livro, como eles são escritos por extenso.

Aproveitando que a tela está limpa, digite a seguinte sequência:

pd 30  
pf 200 (RETURN)

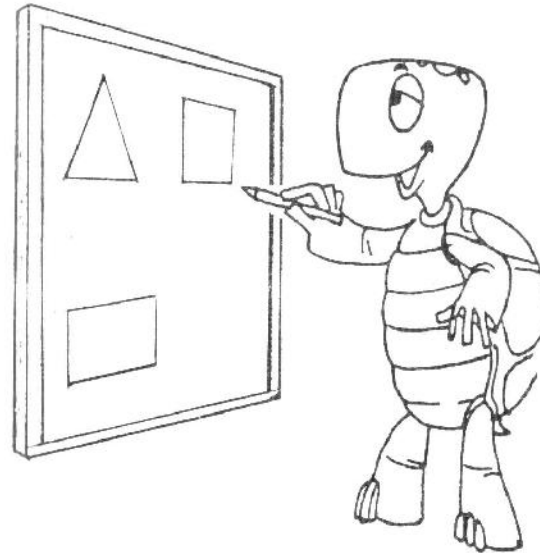
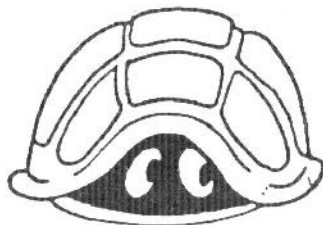
Usando a folha de atividades da pág. 133 do Apêndice, ficará mais fácil entender como é que a tartaruga executa estas instruções.

Com isto você certamente concluiu que a tela de trabalho tem um limite próprio nas direções NORTE-SUL e LESTE-OESTE. Tente traçar um rastro nestas direções fazendo com que sua tela fique igual a que se encontra a seguir. Aproveite o desenho para anotar o quanto é possível andar do centro até a margem superior e do centro até as demais margens.



Observe que estes números vão ser muito importantes para a realização de qualquer trabalho futuro.

O que acha de fazer, na tela do computador, uma cópia do Mapa do Tesouro? Não deixe de anotar, numa folha de papel, todas as instruções que der à tartaruga, uma vez que esta folha será importante para outras atividades propostas no próximo capítulo.



## TRAÇANDO FORMAS SIMPLES

Lembra-se do que falamos sobre a diferença entre os comandos `pf/pt` e `pd/pe`? Como vimos, usamos a expressão `andar` para os comandos `pf` e `pt` e `girar` para os comandos `pd` e `pe`.

Falemos um pouco mais sobre essa expressão "girar".

Os ponteiros do relógio giram, a ventoinha do motor dos carros gira, o "gira-gira" da pracinha, como diz o próprio nome, também gira.

A cada vez que qualquer uma dessas coisas passa pelo mesmo lugar, completa uma volta, não é mesmo?

A tartaruga é igualzinha. Ela gira para a direita ou para a esquerda, dando voltas, como você mesmo poderá observar brincando com a figura contida na pág. 135.

Mantenha o ponteiro maior fixo num ponto qualquer e com o outro vá girando a tartaruga até o próximo tracinho. Continue da mesma forma até que os ponteiros fiquem um sobre o outro. Isto representa que você fez sua companheira girar uma volta completa.

Você talvez já tenha aprendido (mas não custa lembrar) que representamos a medida de um pedaço de volta, ou de uma volta inteirinha, em graus ( $^{\circ}$ ) e que cada volta completa mede  $360^{\circ}$ .

Vamos voltar ao Mapa do Tesouro que você reproduziu na tela e veja o quanto foi preciso fazer sua amiga girar a fim de que a linha traçada do marco 1 ao marco 2 ficasse retinha no sentido horizontal.

O número encontrado foi  $90^{\circ}$ ?

Por que?

Pense bem: se uma volta inteirinha é igual a  $360^{\circ}$ , quanto vale meia volta? Que tal colocar os ponteiros com a tartaruga nesta posição, isto é, dividindo o círculo em duas metades?

Muito fácil, não é mesmo?

Para dar meia volta, é preciso fazê-la girar  $180^{\circ}$ .

Complicando um pouquinho:

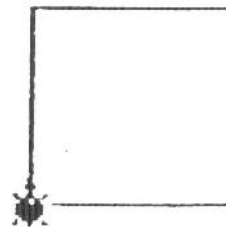
- quanto é o equivalente a metade de meia volta?

Certo,  $90^{\circ}$  é a resposta.

Então, se para fazer o desenho do Mapa você usou o comando **pd** mais a entrada **90**, tudo o que em verdade aconteceu foi que a tartaruga girou a metade de meia volta ou, de maneira mais simples, deu  $1/4$  de volta para a direita.

Ainda é preciso dizer que a cada vez que você desloca o ponteiro menor para um dos tracinhos assinalados, você forma um ângulo.

Depois disso tudo, o que acha de construirmos um quadrado?



Primeiro, vamos pensar juntos:

- o que é um quadrado?

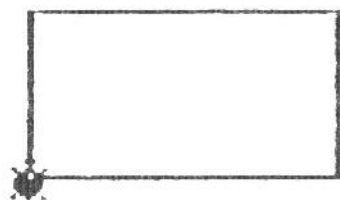
Claro... quadrado é uma figura que tem 4 lados iguais e 4 ângulos retos também iguais.



A proposta então é desenhar esta figura de modo que a tartaruga fique na mesma posição em que estava inicialmente, dando-lhe, para isto, 8 instruções.

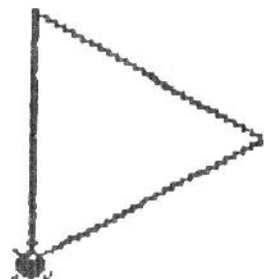
Conseguiu? Ótimo!.

Faça outra figura: o retângulo.



A diferença dele para o quadrado é somente no tamanho dos lados que, no caso do retângulo, são iguais dois a dois. O número de instruções também é o mesmo e a tartaruga deverá terminar na posição inicial.

Como fazer para desenhar um triângulo com três lados iguais?



Numa primeira olhada para a figura, parece que temos um problema: os ângulos agora não são retos! De quanto será a medida destes ângulos?

Vou lhe dar uma dica de grande valor:

- o giro realizado pela tartaruga é sempre pelo lado de fora, ou externo às figuras.

Talvez você esteja pensando:

"E daí?"

Daí, que a soma de todos os giros dados pela tartaruga, para fechar uma figura, tem que ser sempre igual a  $360^\circ$ .

Desta forma, para descobrirmos o ângulo de qualquer figura basta dividir 360 pelo número de lados da figura a ser construída.

Veja os casos do quadrado e do retângulo: ambos têm 4 lados, o que nos leva a conta  $360 \div 4 = 90$ .

Não foi este o giro da tartaruga quando desenhou tais figuras?

Agora ficou fácil fazer o triângulo. Os 3 lados medem o mesmo número de passos e os 3 ângulos medem  $360 \div 3$  ou 120, sendo 6 o número de instruções necessárias.

Repare na sequência de comandos que podemos dar para a construção destas figuras:

QUADRADO

{ pf 30  
pd 90

{ pf 30  
pd 90

{ pf 30  
pd 90

{ pf 30  
pd 90

TRIANGULO

{ pf 50  
pe 120

{ pf 50  
pe 120

{ pf 50  
pe 120

Quantos grupos de instruções iguais você pode contar em cada uma destas figuras?

Com o auxílio de um novo comando - **repita** - podemos fazer as mesmas figuras de maneira bem mais simples.

Para o quadrado, digite:

repita 4 [pf 30 pd 90]

Vamos analisar como a tartaruga compreende esta instrução. O número 4 indica quantas vezes queremos que a LISTA DE COMANDOS que está entre colchetes seja repetida.

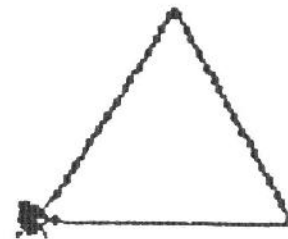


repita é um comando que necessita vir sempre acompanhado de colchetes.

Então, para construirmos um triângulo, basta digitar:

repita 3 [pf 50 pe 120]

Alguma coisa não está boa, pois o triângulo está em pé e gostaríamos que ele ficasse com a base na horizontal, deste jeito:



Para isto, é necessário posicionar a tartaruga antes de construir a figura.

## Experimente:

pd 90  
repita 3 [pf 50 pe 120]

Tente agora fazer um retângulo com o comando **repita**. Uma vez que você já construiu esta figura passo a passo, é fácil encontrar quantos grupos de instruções iguais existem nela.

Anote abaixo como você executou esta tarefa.

Toda vez que você construiu uma nova figura, teve que apagar a anterior ou então deixar que uma ficasse por cima da outra.

É possível pedir à tartaruga que ande pela tela sem deixar nenhuma marca de seu lápis, utilizando o comando **usenada**, ou, de forma simplificada, **un**.

Com seu lápis levantado, podemos levá-la a qualquer ponto da tela e, uma vez estando no lugar desejado, ordenamos **uselápis**, ou **ul**. A tartaruga automaticamente colocará seu lápis pronto para riscar e qualquer instrução dada depois deste comando, ficará registrada na tela.



## USANDO CORES

Você sabia que o HOTLOGO tem 16 cores e que com elas você pode fazer um milhão de coisas diferentes como mudar a cor do fundo da tela, a cor da tartaruga ...

Isso mesmo!

Já pensou como ficará simpática nossa amiguinha em vermelho?

Vamos por partes:

- primeiro, a relação das cores:

0 transparente	8 vermelho
1 preto	9 vermelho claro
2 verde	10 amarelo
3 verde claro	11 amarelo claro
4 azul escuro	12 verde escuro
5 azul	13 rosa
6 vermelho escuro	14 cinza
7 azul claro	15 branco



*Consulte a relação de cores quando estiver realizando alguma atividade que a necessite.*

Ah! Você agora quer saber como fazer a tartaruga mudar de cor? Simples. Experimente o comando:

mudect 8

Pronto! Aí está a nossa amiga em vermelho. Sugiro que você ande com ela pela tela.

O que aconteceu?

Se você esperava um traço vermelho, isto é, da mesma cor da tartaruga, deve ter ficado surpreso, pois ela continuou deixando um rastro branco.

Calma! Claro que também podemos mudar a cor do lápis usado por nossa amiga.

Para um traço vermelho você terá então que usar

mudecl 8

Brinque um pouco com estes dois comandos, usando outros números e fazendo assim com que a tartaruga e seu traço fiquem de várias cores.

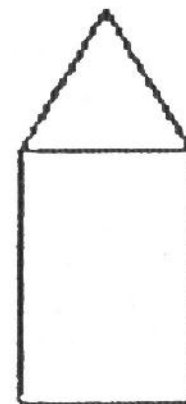
Gostoso, não? Mas ainda é pouco! O que acha de mudarmos também a cor do fundo de nossa tela? Para isso digite:

mudecf ---

colocando no lugar de --- um dos números da lista de cores. Tenha o cuidado de não escolher para o fundo a mesma cor da tartaruga e de seu traço pois então você não poderá vê-los.

Imagino que sua criatividade deva estar a mil, mas podemos fazer coisas muito mais legais.

Olhe para esta figura:



Certamente você pensou: "Já fiz coisas muito melhores!"

Mas, e se cada parte dela fosse preenchida com uma cor? Sua opinião ainda é a mesma ou já lhe parece mais interessante?

Vamos, então, construir nossa figura da seguinte maneira:

```
repita 2 [pf 60 pd 90 pf 40 pd 90]
pf 60
pd 30
repita 3 [pf 40 pd 120]
```

Estando ela pronta, escolha uma cor para pintar o telhado e outra para a parede. Nossa! Como você é rápido!

Quando queremos pintar um desenho não podemos esquecer de um pequeno mas importante detalhe:

- a tartaruga não pode, em nenhuma hipótese, estar sobre seu traço.

Desta forma, para pintarmos primeiramente o telhado de nossa casa, precisamos levar a tartaruga para o seu interior, sem usar o lápis, através das seguintes instruções:

```
un
pd 30
pf 10
```

Até aí, tudo bem?

Você já sabe que para deixar um rastro a tartaruga precisa estar usando o lápis, o que também é necessário para pintar qualquer figura.

Digite, para finalizar, os comandos:

```
ul
mudecl ... (coloque aqui o
              número
              escolhido)

pinte
```

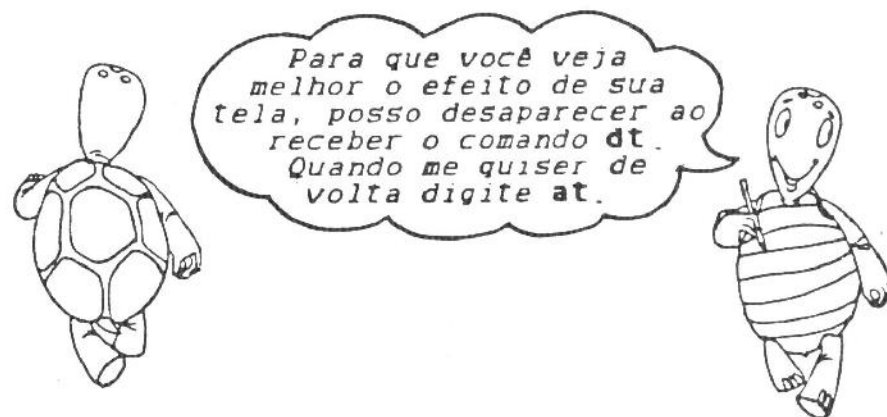
Que tal? O visual é bonito? Tente você mesmo pintar a parede, anotando aqui suas instruções:

---

---

---

---



Apenas mais um detalhe:

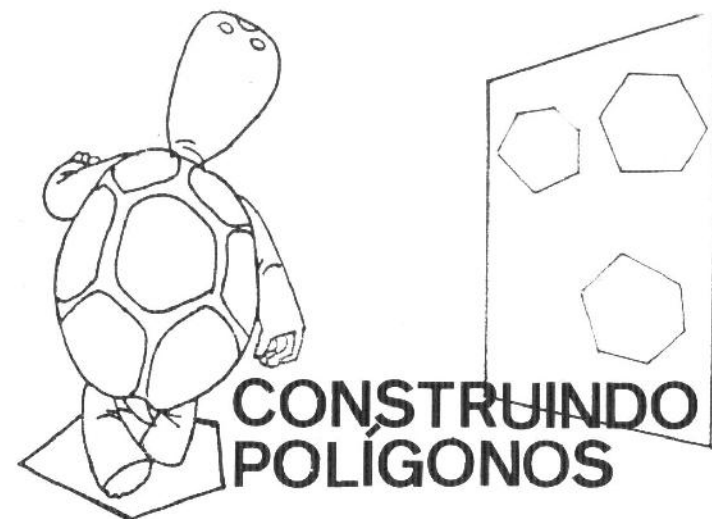
- digite a sequência

```
pf 45 pd 120 pf 45 pd 120
pf 43 pd 120
un pf 10 dt
ul pinte
```

Ficou surpreso com o que aconteceu?

É exatamente para este detalhe que você precisa ficar atento. Toda vez que formos usar o comando **pinte**, o desenho a ser pintado deve estar completamente fechado pois do contrário a cor vazará e preencherá a tela toda.

Libere de vez toda a sua criatividade e elabore, usando estes novos comandos e recursos, alguns projetos super-coloridos.



Como podemos definir o que é um polígono?

Qualquer figura fechada por linhas retas, sem passar duas vezes pelo mesmo ponto, é um polígono.

Existem dois tipos de polígonos:

1) os regulares, que são figuras com lados e ângulos iguais - como o quadrado e o triângulo equilátero;

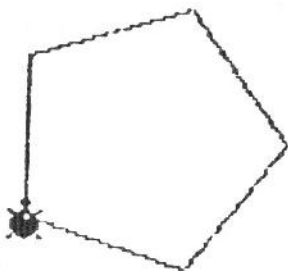
2) e os irregulares com lados e/ou ângulos diferentes - como o retângulo.

Como já conversamos no Capítulo 3, para construir qualquer polígono regular encontramos o ângulo a ele correspondente através da divisão do giro total da tartaruga ( $360^\circ$ ) pelo número de lados da figura que você deseja executar.

Por exemplo: para um polígono de 5 lados (pentágono) dividimos 360 por 5, para achar 72, que é o valor do ângulo desta figura.

Usando o comando `repita`, teríamos:

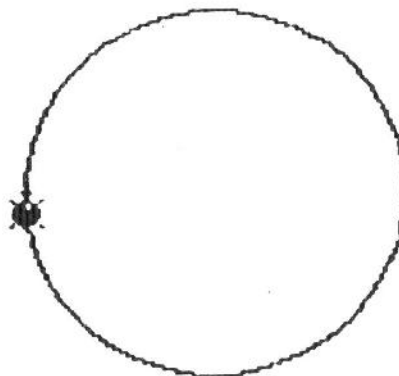
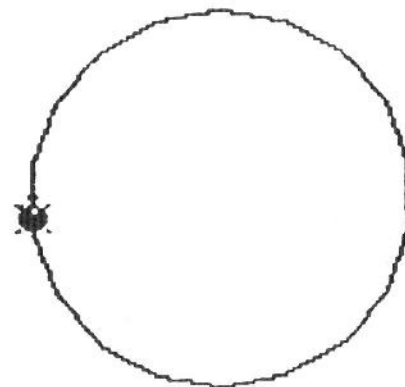
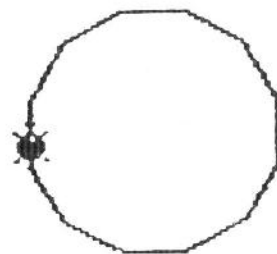
```
repita 5 [pf 30 pd 72]
```



Fica muito fácil, a partir daqui, a construção de outros polígonos, como os de 6, 8, 10, 12, 18, 36 lados, etc...

Experimente construí-los e observe o que vai acontecendo.

Você percebeu que quanto maior é o número de lados, menor é o valor correspondente ao ângulo e que a figura está se tornando cada vez mais arredondada?



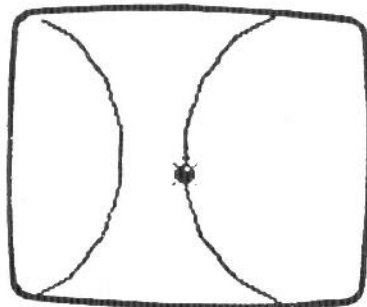
A partir desta observação, que instrução você acha mais adequada para construir um polígono semelhante a uma circunferência?

Isso mesmo! A resposta é:

```
repita 360 [pf 1 pd 1]
```

pois o polígono mais parecido com uma circunferência é o de 360 lados que divididos por 360 resulta em 1 de ângulo.

Acontecendo da figura "estourar" o tamanho da tela,



você poderá diminuí-la reduzindo o tamanho do passo da tartaruga.

Como fazer para encontrar um número menor que 1 ? Basta que se use os decimais.

Explore o círculo em vários tamanhos, usando a alternativa dos decimais, com instruções como esta:

```
repita 360 [pf 0,5 pd 1]
```

Aproveite os comandos aprendidos no capítulo sobre o uso das cores para encher sua tela de figuras bem coloridas.



*Todas estas figuras podem ser usadas em seus projetos.*

Existem ainda duas outras figuras que poderão ser utilizadas por você: o trapézio e o paralelogramo.

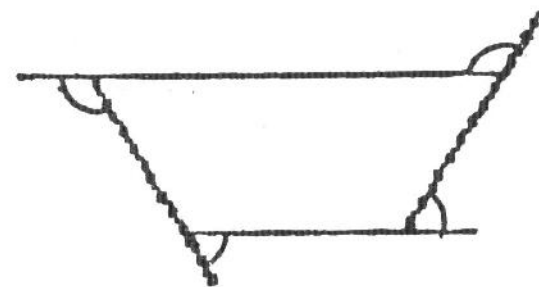
Como dissemos no início deste Capítulo todas duas possuem as características dos polígonos irregulares: lados e/ou ângulos diferentes.

Para desenhar um trapézio não é possível usar comando **repita**, sendo suas instruções:

```
pd 90 pf 40 pe 60
```

```
pf 25 pe 120 pf 65
```

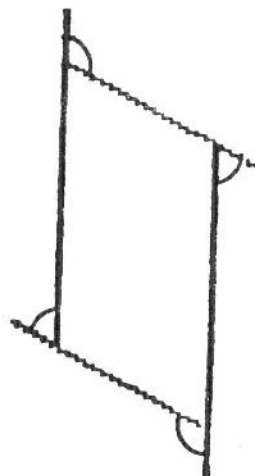
```
pe 120 pf 25
```





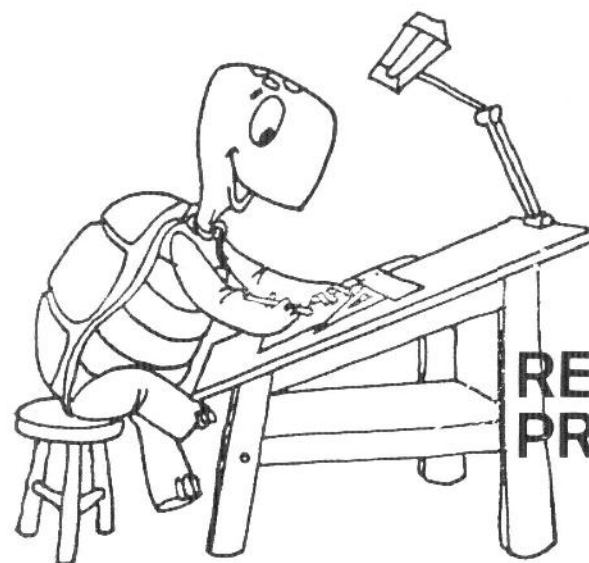
Já para fazer um paralelogramo, usamos:

repita 2 [pf 70 pd 120 pf 40 pd 60]



O paralelogramo pode ser usado para representar o telhado de uma casa. No entanto, é preciso, como já foi feito com o triângulo, posicionar a tartaruga adequadamente antes de desenhá-lo.

Repare que nestas duas últimas figuras, a soma dos giros é sempre igual a  $360^\circ$ .



## REALIZANDO PROJETOS

Aqui você encontrará vários desenhos. São apenas sugestões, mas você deve tentar executar pelo menos dois deles, porque isto facilitará o seu desenvolvimento e aprendizado nas próximas etapas deste livro.

Não se esqueça de utilizar todos os comandos e recursos aprendidos até aqui e de anotar todas as instruções dadas numa folha de papel, pois elas sempre terão utilidade.

Antes de começar, vamos aprender um novo comando: **useborracha** ou **ub**.

Como o nome diz, ele faz com que a tartaruga troque seu lápis por uma borracha, permitindo assim apagar qualquer rastro que não esteja exatamente como você quer, desde que passe por cima dele.

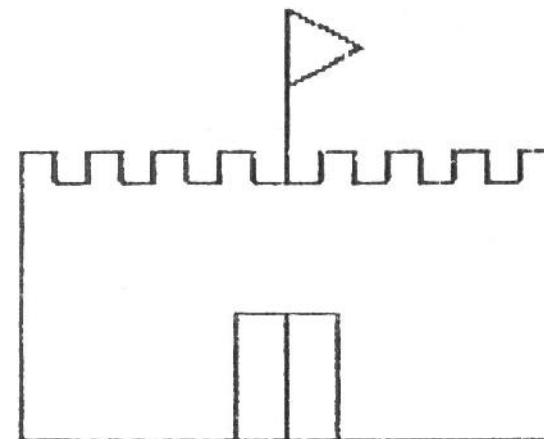
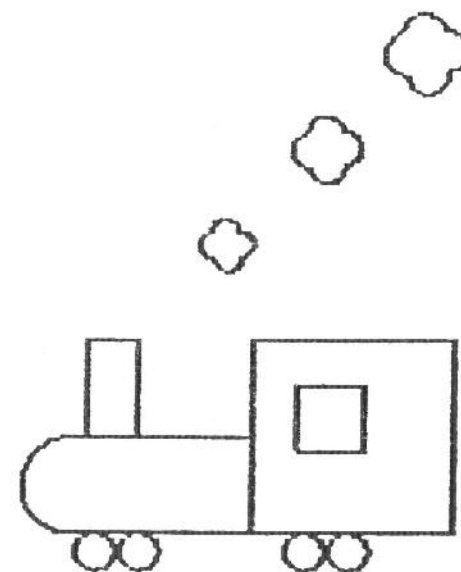
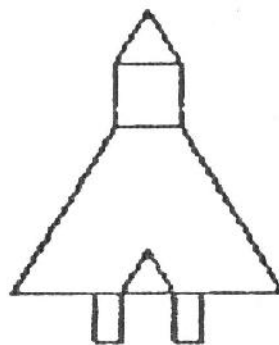
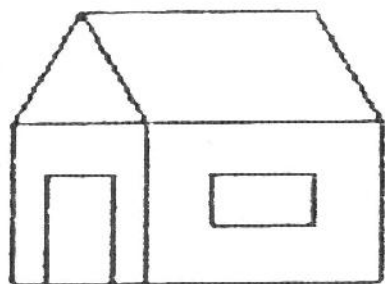
Para ela voltar a riscar peça  
uselápis ou ul.

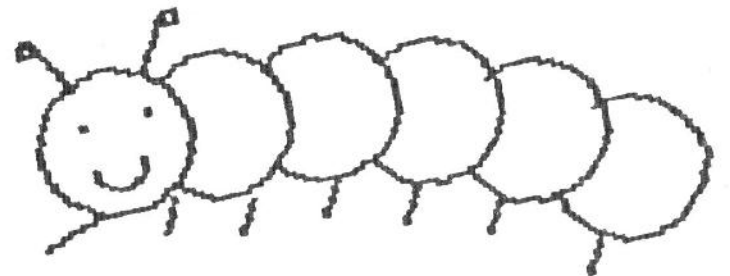
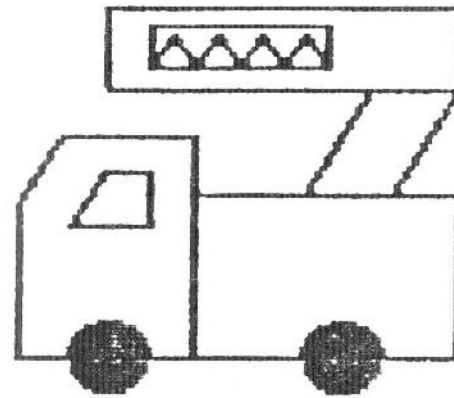
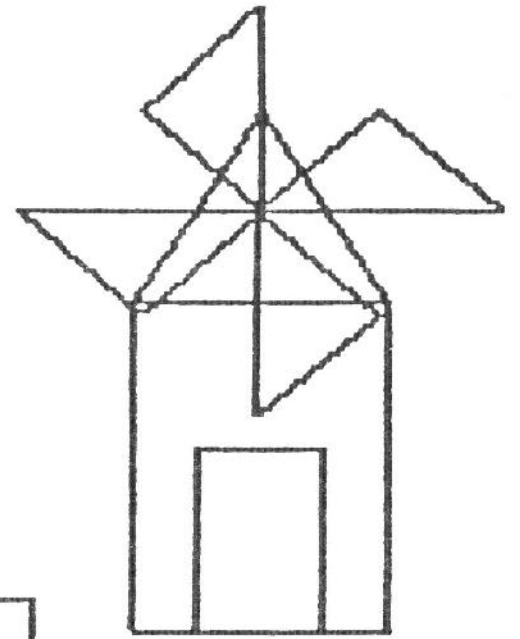
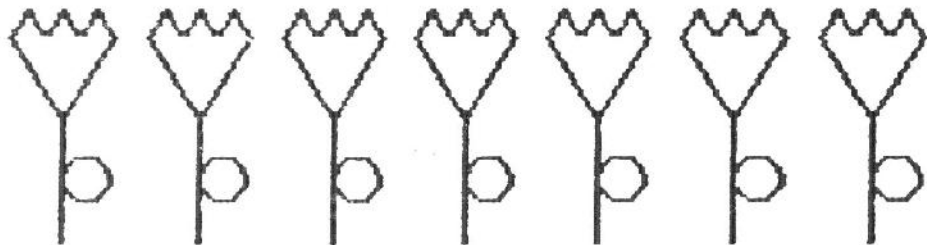
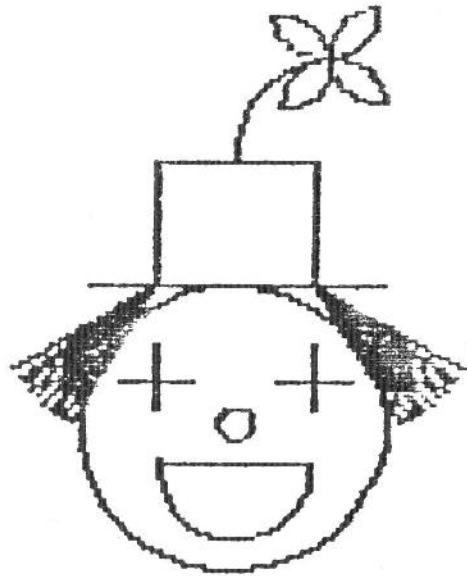
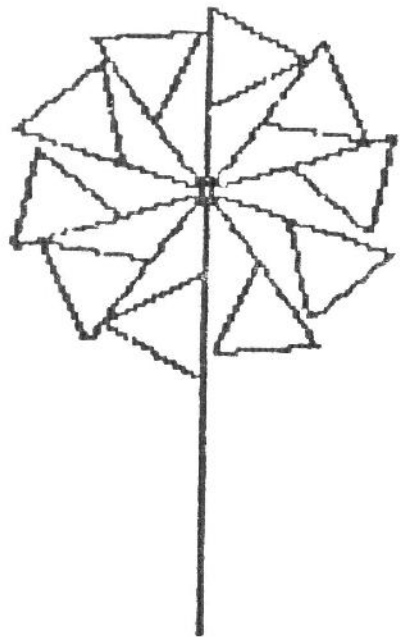
Exemplo:

pf 40 ub (e RETURN)

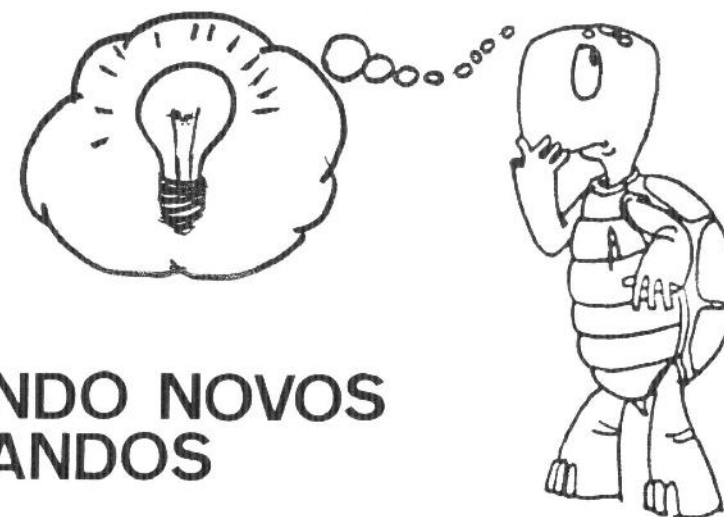
pt 40 ul (e RETURN)

Agora você já está pronto para  
construir seus primeiros projetos,  
podendo "consertar" aquilo que não  
sair da mesma maneira que imaginou.





Um dia você acorda e descobre que a Terra está sendo invadida por seres extra-terrestres. Desenhe nesta folha suas naves e como são esses seres. Em seguida tente reproduzi-la no computador.



## CRIANDO NOVOS COMANDOS

O grande barato deste capítulo é que a partir daqui você vai poder ensinar ao computador PROCEDIMENTOS que serão entendidos e executados pela tartaruga.

Primeiro é preciso saber o que é um **PROCEDIMENTO**.

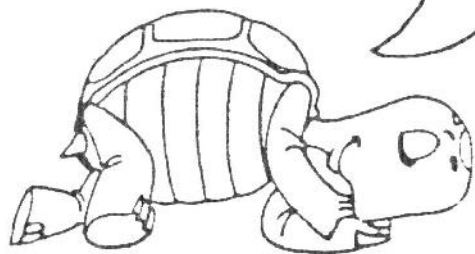
Escrever um procedimento tem o mesmo significado de escrever um programa, isto é, colocar em ordem uma sequência de instruções que, por enquanto, resultarão num desenho.

Até então você esteve trabalhando com o computador através do MODO DIRETO, o que quer dizer que para desenhar qualquer figura você precisava digitar instrução a instrução.



Com o comando aprenda você passa do MODO DIRETO para o MODO DE DEFINIÇÃO, quando o HOTLOGO fica aguardando que você lhe ensine alguma coisa.

*Toda vez que você estiver definindo um procedimento eu estarei descansando, não podendo executar nada que me peça.*



Digitando

ap retângulo

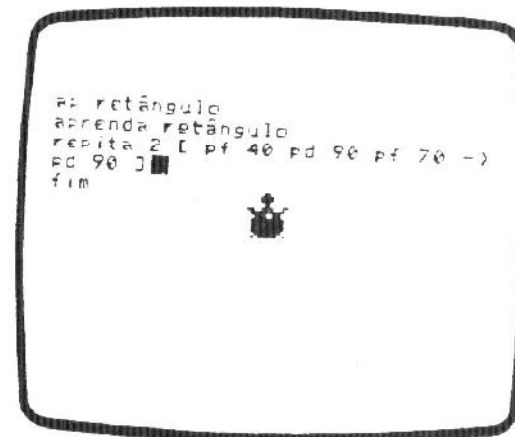
sua tela ficará assim:



Observe que entre as duas últimas linhas não há o espaço necessário para colocar as instruções que permitirão desenhar o retângulo.

Este espaço pode vir sendo "aberto", linha a linha, sempre que você pressionar a tecla RETURN.

Faça isto e acrescente então as instruções de um retângulo, como mostra a figura abaixo:



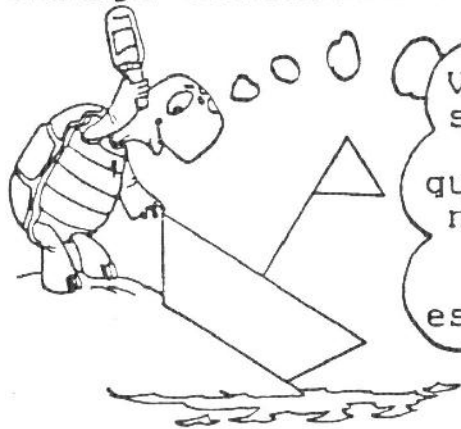
Para avisar ao HOTLOGO que o procedimento já está definido, pressionamos a tecla ESC e imediatamente veremos surgir a mensagem:

retângulo aprendido

Mas, preste atenção... Definir um procedimento não quer dizer que o mesmo esteja correto, pois isto depende das instruções dadas e também que estas tenham sido digitadas sem erro.

Daqui prá frente a palavra **retângulo** torna-se um comando conhecido pelo HOTLOGO podendo ser usada em outros procedimentos tão naturalmente quanto os comandos **pf**, **pt**, **pd**, **tat**, etc ...

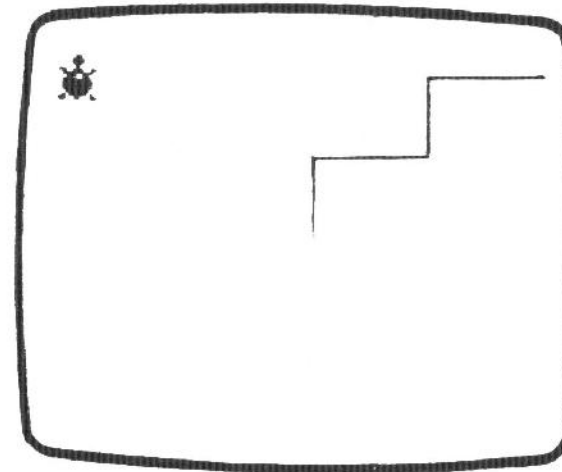
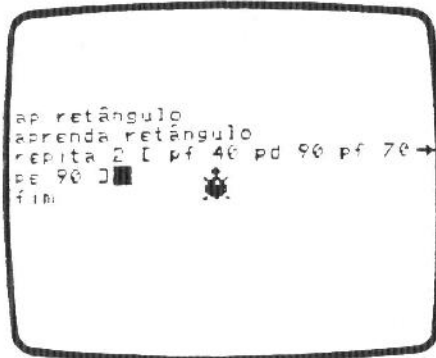
Sempre que você a digitar a tartaruga desenhará esta figura.



Você pode "batizar" seus procedimentos com qualquer nome que a tartaruga ainda não conheça. Assim os novos comandos serão criados e escolhidos por você.

Há pouco falamos em instruções e digitação corretas. Pode acontecer de você cometer erros como os exemplos a seguir.

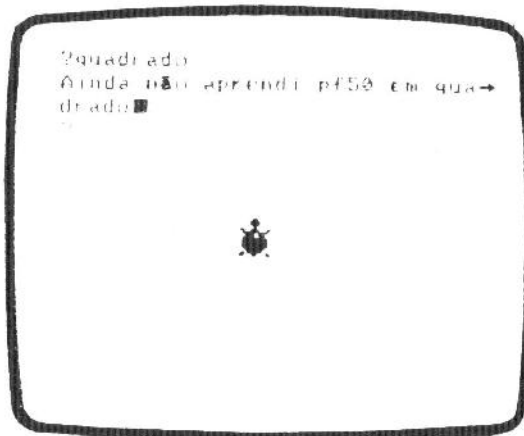
Acompanhe comigo, digitando-os:



Onde está o erro?

Os dois giros deveriam ser **pd** ou **pe**.





Neste caso, a tartaruga já disse o que há de errado no procedimento. Será que dá para consertar ou é preciso ensinar tudo outra vez?

Não desanime! É muito simples corrigir estes erros.

Como? Conhecendo um novo comando: **ed** que deve vir seguido do nome do procedimento a ser corrigido, dentro de colchetes. Proceda assim:

```
ed [quadrado]
```

A partir do momento que a tecla **RETURN** for pressionada, o **HOTLOGO** listará o procedimento pedido.

Efetue agora os consertos necessários com o auxílio das seguintes teclas de correção; não esquecendo de pressionar **ESC** ao final das mesmas.

- ▶ - leva o cursor para a direita.
- ◀ - leva o cursor para a esquerda.
- ▼ - o cursor desce uma linha.
- ▲ - o cursor sobe uma linha.

**CTRL+K** - deleta (apaga) a linha a frente do cursor.

Você sabia que, enquanto o computador estiver ligado, vai guardando - num lugarzinho da memória - todos os procedimentos que lhe são ensinados através do comando **ap**?

Você também sabia que no momento em que for desligado, ele "esquece" tudo?

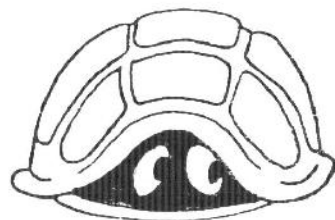
Além das teclas vistas acima, fora do modo de edição você pode usar também as seguintes combinações:

**CTRL+A** - desloca o cursor para o início da linha sem apagá-la.

**CTRL+H** - igual ao ◀◀.

**CTRL+I** - insere uma linha de espaço.

- CTRL+M** - igual ao RETURN.
- CTRL+Y** - repete a última linha digitada.



## GERANDO ARQUIVOS

Que sensação você experimentou ao desligar o computador depois de tanto trabalho para ensinar ao HOT-LOGO novos comandos?

Sei que na maioria das vezes esta sensação é de desânimo, pois é duro pensar que o trabalho terá de ser refeito na próxima vez em que formos lidar com a tartaruga.

Responda com sinceridade:

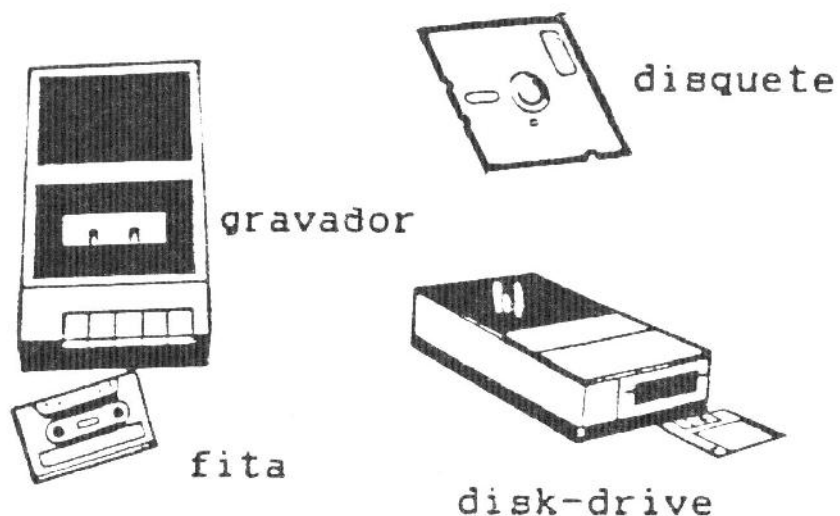
Você consegue lembrar o telefone de todos os seus amigos?

Em muitos casos eles são tantos que precisamos usar uma agenda telefônica para guardá-los.

No LOGO é a mesma coisa. Podemos criar tantos procedimentos que seria impossível memorizá-los, tornando-se necessário que tenhamos um lugar especial para guardá-los.



Este lugar é conhecido, em computação, como memória auxiliar e, tanto pode ser uma fita cassete como um disco.



Como fazer para gravar?

Existem alguns passos importantes a serem seguidos:

- 1) Conectar o gravador ao micro.
- 2) Pôr na memória do computador, através do comando **aprenda**, todos os procedimentos que você deseja gravar.
- 3) Digitar o comando **mots** que nos permite ter acesso aos títulos de todos os procedimentos ensinados, para verificar se não esquecemos nenhum.

- 4) Digitar o comando gravetudo "projetos para o gravador, ou gravetudo "trabalhos para o drive e teclar **RETURN** para iniciar a gravação.



*Coisas importantes:*  
1 - Não esqueça de deixar um espaço entre o comando e as aspas.  
2 - Nunca feche as aspas.

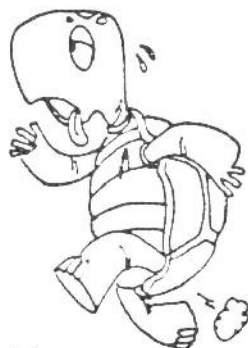
A partir deste momento você gerou um arquivo chamado **projetos** ou **trabalhos**, onde estão contidos todos os procedimentos que haviam sido passados para a memória do computador.

Vejamos isto de forma mais prática:

- gravei um arquivo com o nome **saca.rolha**, onde estão guardados estes procedimentos:

- corsário
- boneca
- nave.espacial
- boneco.de.neve

Ufa! Já dá para desligar o computador sem perder todo o trabalho.



Quando quisermos realizar a operação inversa - que é passar o conteúdo de sua memória auxiliar para a do computador - procedemos assim:

- se estiver usando um gravador, o comando adequado é

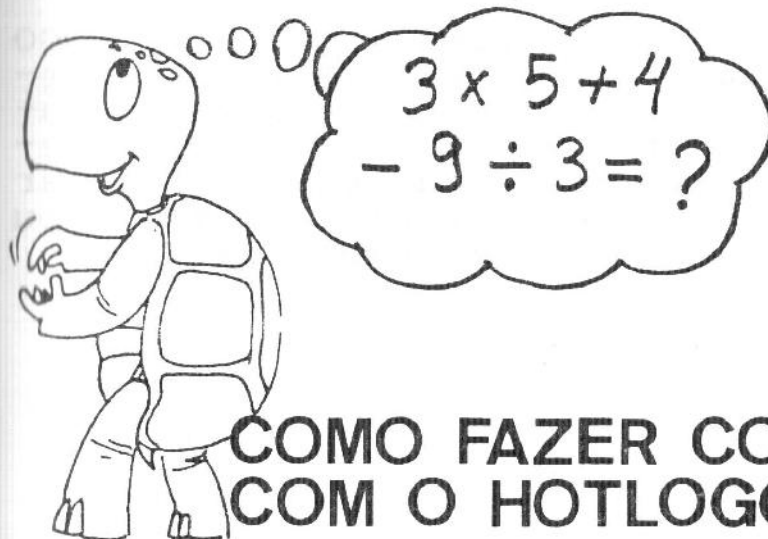
carreguec "saca.rolha

- se for um drive, o comando é parecido com o acima, apenas sem o "c" final que indica a utilização de um cassete:

carregue "saca.rolha



Olhe no manual de instruções de seu micro, como operar corretamente um gravador para gerar arquivos.



## COMO FAZER CONTAS COM O HOTLOGO

Gostaríamos de fazer, neste capítulo, algumas observações sobre como o HOTLOGO opera com os números.

As quatro operações são simbolizadas por:

- \* -> multiplicação
- / -> divisão
- + -> adição
- -> subtração

Para efetuar essas operações utilizamos o comando `escreva` ( ou `esc`, ), desta forma:

```
esc 5 + 7 * 2
```

```
----> 19
```

Como na matemática, o HOTLOGO obedece a hierarquia de sinais, efetuando primeiro as operações indicadas pelos sinais fortes (multiplicação e divisão) para depois executar as que têm sinais fracos (adição ou subtração), como nos exemplos:

```
esc 20 / 4 * 3 + 6
```

```
--> 21
```

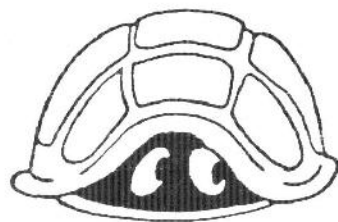
```
esc 36 - 6 * 5 / 3 + 11
```

```
--> 15
```

Caso você queira alterar esta ordem, faça uso dos parênteses:

```
esc (5 + 7) * 2
```

```
--> 24
```



## APRENDENDO VARIÁVEIS

Em algum momento aconteceu de você querer construir a mesma figura, como por exemplo o quadrado, de vários tamanhos?

E como você agiu diante desta situação? Provavelmente através de uma destas opções:

- 1) Chamou **quadrado** e o alterou para o tamanho desejado.
- 2) Ensinou um novo procedimento, para o qual teve que dar outro nome, apenas aumentando ou diminuindo os lados da figura.

Observe com atenção:

```
repita 4 [pf 30 pd 90]
```

```
repita 4 [pf 60 pd 90]
```

Agora responda:

- qual a diferença encontrada nestas duas instruções?

Sem muita dificuldade o que se pode concluir é que as duas são quase iguais. A única coisa que **VARIOU** foi o valor da entrada para o comando **pf**.

É possível criar um único procedimento que permita construir uma figura em diferentes tamanhos.

Ele é assim:

```
ap quadrado :lado
repita 4 [pf :lado pd 90]
fim
```

*Podemos usar outros nomes em lugar de lado. O importante é que antes deles você use (:) dois pontos*



O que significa a expressão **:lado** ?

Ela avisa ao HOTLOGO que ali, através do teclado, vai ser colocado um número por você escolhido, determinando o tamanho do LADO de sua figura.

Como é que você vai usar, então, este novo procedimento?

Depois de passá-lo para a memória do computador, experimente pedir **quadrado 50** e acompanhe comigo como isto funciona dentro do HOTLOGO.

Ao ler **quadrado 50** a tartaruga entenderá que este número é a entrada necessária para substituir a expressão **:lado** usada junto ao comando **pf**, permitindo, assim, a construção de um quadrado de lado 50.

A essa entrada, que pode ter diferentes valores, damos o nome de **VARIÁVEL**.

Explore um pouco este novo recurso, elaborando procedimentos variáveis para trabalhar com figuras como o triângulo, o círculo, etc.

Para figuras como o retângulo e o paralelogramo, que têm lados iguais dois a dois, precisamos de um procedimento que aceite duas entradas:

```
ap ret :lado1 :lado2
repita 2 [pf :lado1 pe 90
          pf :lado2 pe 90]
fim
```

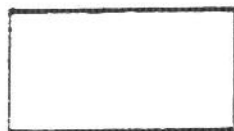
*Digite as instruções entre colchetes numa linha só, sempre que as encontrar impressas como neste procedimento.*



Consequentemente, para executá-lo, precisamos fornecer dois números:



ret 70 40



ret 40 70

Complicando um pouquinho:

- o que muda quando construímos um triângulo, um quadrado e um pentágono?

Para facilitar seu raciocínio, observe:

```
repita 3 [pf 40 pd 120]
```

```
repita 4 [pf 40 pd 90]
```

```
repita 5 [pf 40 pd 72]
```

Nestes casos, há duas diferenças, que estão intimamente ligadas: a primeira é o número de lados de cada figura (3, 4 e 5); a segunda é o ângulo delas (120°, 90° e 72°).

Já vimos, em TRAÇANDO FORMAS SIMPLES, que o ângulo é sempre resultante da divisão de 360° (giro total da tartaruga) pelo NUMERO DE LADOS da figura.

Dáí podemos escrever um procedimento para fazer figuras diferentes, como este:

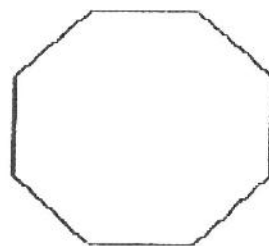
```
ap figuras :nlados  
repita :nlados [pf 40  
pd 360/:nlados]  
fim
```

Use figuras 4

figuras 6

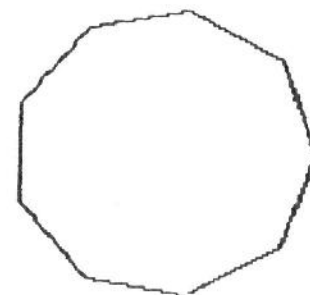
figuras 7

ou qualquer outro número que você ache legal.

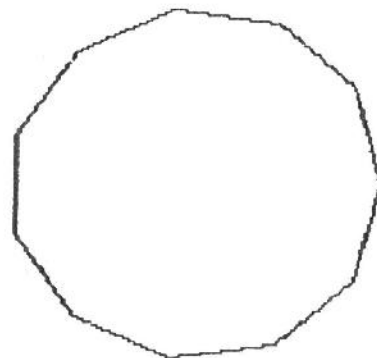


figuras 8

figuras 9



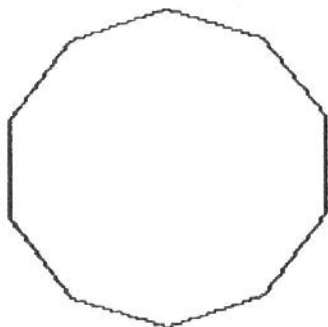
figuras 11



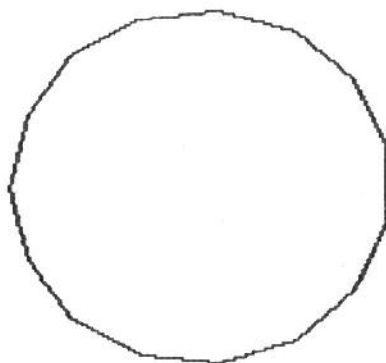
Bom, se pudemos mexer no tamanho das figuras e em seu número de lados, podemos também fazer as duas coisas ao mesmo tempo:

```
ap figuras :lado :nlados  
repita :nlados [pf :lado  
pe 360/:nlados]  
fim
```

Veja como conseguimos estas figuras:

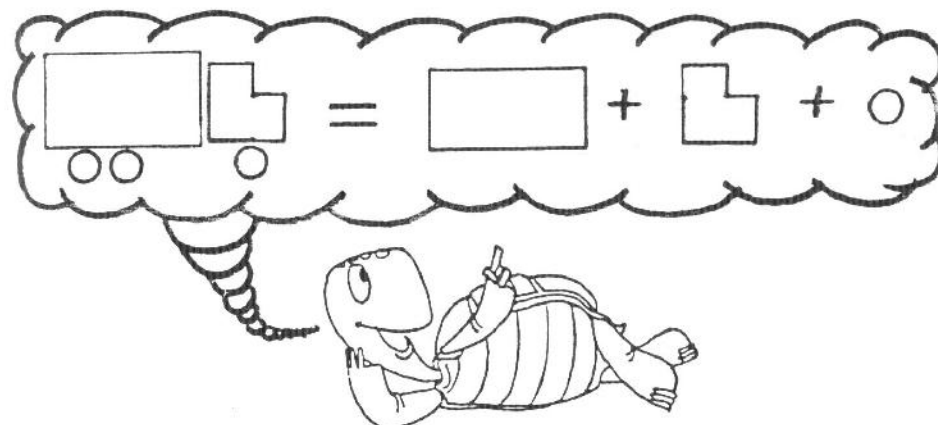


figuras 40 10



figuras 30 15

## DECOMPONDO UM PROJETO EM PARTES

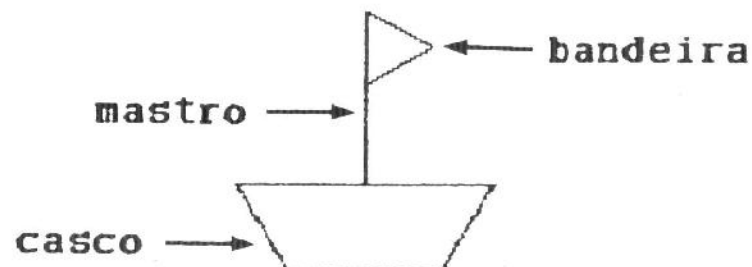


Quando você pensa num desenho, a imagem que se forma é dele pronto, com todos os detalhes.

Na hora de passar essa idéia para o papel ou para o computador, não é possível fazer tudo ao mesmo tempo.

É preciso ir traçando uma parte de cada vez, até chegar ao ponto que você imaginou.

Observe a figura do barco e veja do que ele é composto:



Podemos construir procedimentos específicos para cada uma destas partes, como estes:

```
ap casco
pd 90 pf 40 pe 60 pf 30
pe 120 pf 70 pe 120 pf 30
fim
```

```
ap mastro
pf 45
fim
```

```
ap bandeira
repita 3 [pf 30 pd 120]
fim
```

e juntá-los num único procedimento chamado barco.

```
ap barco
casco
mastro
bandeira
fim
```

Desenhe aqui o resultado obtido ao chamar **barco**.

O que será que aconteceu?

Você já deve ter concluído que ao terminar de traçar o casco a tartaruga não foi posicionada de forma conveniente para traçar o mastro e a bandeira.

O procedimento com a alteração necessária é:

```
ap barco
casco
pe 60 pf 20 pe 90 un
pf 27 ul
mastro
bandeira
fim
```

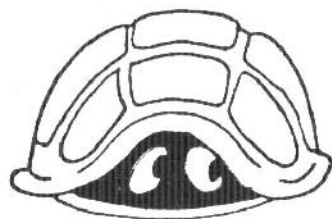
**casco, mastro e bandeira**, quando usados dentro do procedimento **barco**, passaram a ser chamados de subprocedimentos.



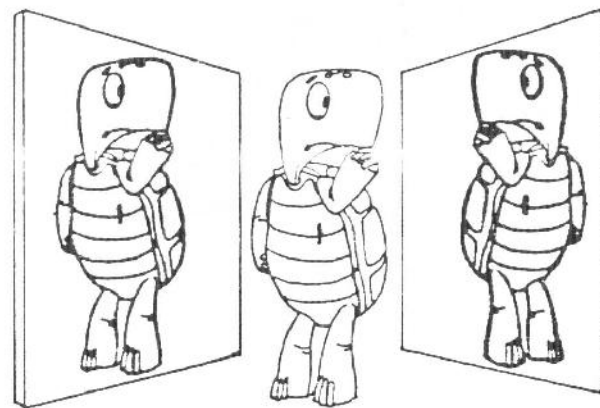
*Quando necessário,  
não esqueça de me  
posicionar corretamente  
entre dois  
subprocedimentos.*

Que tal agora usar barco como subprocedimento de paisagem?

O resto do projeto fica por conta de sua imaginação.



## ELABORANDO PROCEDIMENTOS RECURSIVOS



*Procedimentos recursivos ...*

*Recursão... O que é isto?*

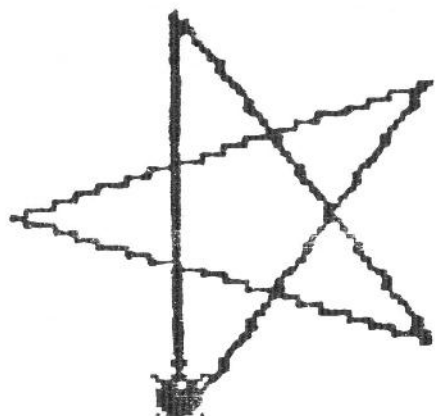
Antes de escrever uma definição, vamos ver o que acontece na prática: pegue dois espelhos, coloque um na sua frente e o outro nas costas.

O que você verá então é a sua imagem refletida de frente e de costas, alternadamente, e reproduzida infinitas vezes.

Veja agora como é um procedimento recursivo. Digite:

```
ap estrela  
repita 5 [pf 30 pe 144]  
estrela  
fim
```





O que faz este procedimento?

Em sua primeira linha dá as instruções para a construção de uma estrela. Na linha seguinte está escrito novamente o nome do procedimento e isto faz com que ele execute outra vez a estrela.

Assim como a experiência do espelho, o procedimento `estrela` se repete indefinidamente, pois está sempre chamando a si mesmo, ocasionando então a **RECURSAO**.

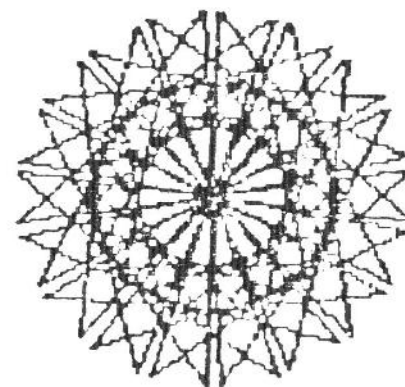
Nestes casos, a única forma de interromper a execução do procedimento e apertando as teclas **CTRL** e **STOP**, simultaneamente.

Será que se introduzirmos uma pequena alteração em `estrela` não conseguiremos obter um visual diferente?

Tente, por exemplo, acrescentar esta linha:

```
ap estrela
repita 5 [pf 30 pe 144]
pe 20
estrela
fim
```

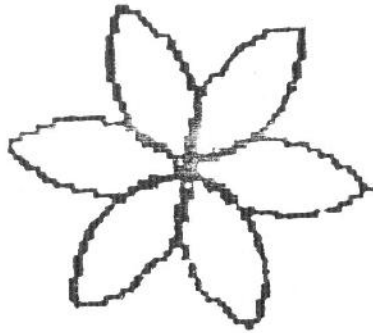
O resultado aqui obtido ficou diferente do anterior, não é?



Vamos experimentar outros procedimentos com recursão:

```
ap flor :n
pétala :n
pd 30
flor :n
fim
```

```
ap pétala :n
repita 9 [pf :n pd 10]
pd 90
repita 9 [pf :n pd 10]
fim
```

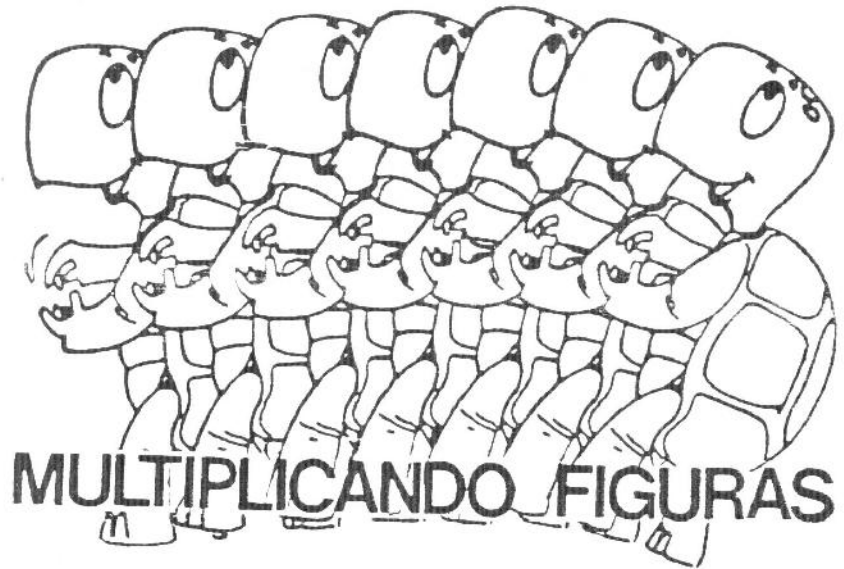
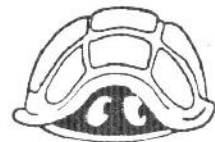


flor 4

pétala 4



Daqui para a frente fica por sua conta criar figuras super interessantes com o auxílio deste novo recurso.



## MULTIPLICANDO FIGURAS

Até aqui, você aprendeu diversos comandos assim como alguns recursos, que lhe permitiram obter vários desenhos com o auxílio da tartaruga.

O que você acharia de poder trabalhar com mais de uma tartaruga?

Seria genial, não é mesmo?

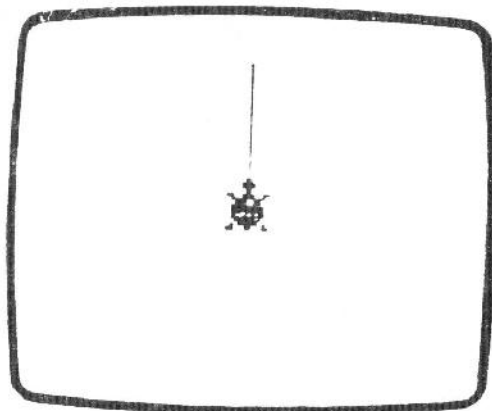
Com o HOTLOGO podemos chamar até 30 delas, mas antes disso é importante saber de alguns detalhes.

A tartaruga que estamos acostumados a trabalhar é a de número 0.

Para chamar outra qualquer, basta digitar o comando `atat`, que significa atenção tartaruga, seguido de um número entre 0 e 29.

Vamos experimentar?

atat 1 pf 30



Olhando para seu vídeo, você somente vê a tartaruga original no centro da tela e um risco de 30 passos.

Onde estará então a tartaruga 1?

Na verdade ela está na ponta deste traço. Só que para vê-la é necessário digitar at (apareça tartaruga).

De agora em diante toda nova instrução só será "ouvida" por esta tartaruga e assim conseqüentemente, ou seja, caberá à última tartaruga acionada executar as últimas instruções.



Calma!

É possível movimentar várias delas simultaneamente, mas para isso precisamos usar o comando rg que desativa todas as tartarugas acionadas anteriormente.

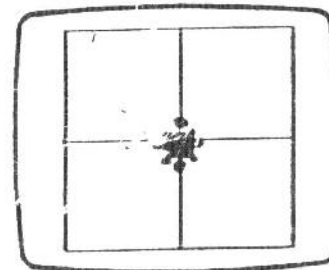
Faça isso e experimente depois digitar esta sequência de instruções:

```
atat 1 pd 90
atat 2 pd 180
atat 3 pd 270
atat [0 1 2 3] at
repita 4 [pf 20 pd 90]
```



Quando você quiser acionar, ao mesmo tempo, mais de uma tartaruga, coloque os números correspondentes entre colchetes.

Você conseguiu assim obter quatro figuras desenhadas ao mesmo tempo.



Não é só isso!

Existe ainda um comando que dá velocidade às tartarugas ativadas e pode ser alterado mesmo que elas estejam em movimento:

`mudevel` (seguido de um número entre -128 e 128)

Experimente:

`mudevel 30` e, em seguida,  
`mudevel -78` .

Como você viu, os números negativos (com o sinal de menos antes deles) indicam que a tartaruga deve andar para trás.

E para fazer a tartaruga parar?

Há duas maneiras:

`mudevel 0`

que é velocidade nula, se ainda queremos continuar trabalhando com estas tartarugas ou,

`rg`

se quisermos pará-las e desativá-las.

Faça um procedimento, dando um nome por você escolhido, com as instruções a seguir e veja que efeito bonito obterá na tela.

```
dt
atat [ 1  2  3  4  5  6  7  8  9
      10 11 12 13 14 15 16 17 18
      19 20 21 22 23 23 24 25 26
      27 28 29]
```

```
at un
cada [pf quem * 6]
pd 90
cada [mudect quem + 1]
mudecf 1
cada [mudevel quem * 4]
```

Observe agora, atentamente, todos os passos dados para a construção deste procedimento:

- 1ª linha: utilizamos o comando `dt` para fazer desaparecer a tartaruga de número 0.
- 2ª linha: acionamos todas as tartarugas (de 1 a 29).
- 3ª linha: com os comandos `at` e `un` pedimos para todas aparecerem e levantarem seus lápis.
- 4ª linha: aqui você encontrou dois novos comandos que serão explicados separadamente.

`cada` - faz com que cada uma das tartarugas ativadas execute uma sequência de instruções. Por isso a necessidade de usar os colchetes.

quem - devolve os números das tartarugas ativadas, no nosso caso de 1 a 29.

Explicando melhor a linha,

cada [pf quem \* 6]

```
tartaruga 1 anda pf 1 * 6
tartaruga 2 anda pf 2 * 6
tartaruga 3 anda pf 3 * 6
```

```
.....
.....
.....
.....
.....
```

```
tartaruga 27 anda pf 27 * 6
tartaruga 28 anda pf 28 * 6
tartaruga 29 anda pf 29 * 6
```

O resultado destas multiplicações servem de entrada para o comando pf.

5a linha: giramos 90° para a direita

6a linha: funciona como a 4a linha sendo o resultado das adições a entrada necessária para o comando mudect.

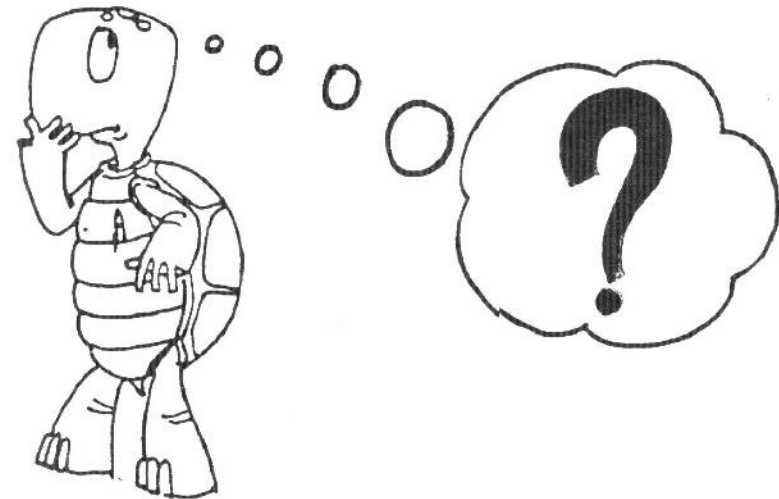
7a linha: mudamos a cor do fundo para melhor visualizar as cores das tartarugas.

8a linha: funciona como as linhas 4 e 6, sendo que o resultado das multiplicações serviu de entrada para o comando mudlevel.

Você pode agora alterar este procedimento como achar melhor para obter novos efeitos visuais.

Antes, porém, responda:

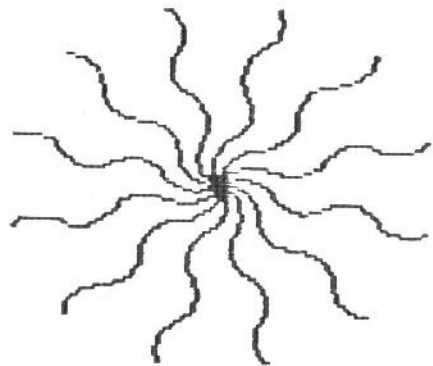
Por que desativamos a tartaruga número 0?



Experimente ensinar os dois procedimentos abaixo e veja que bonito visual obterá:

```
ap raio
repita 4 [ repita 9 [ pf 1 pd 10 ]
repita 9 [ pf 1 pe 10 ] ]
fim
```

```
ap sol
atat [ 0 1 2 3 4 5 6 7 8 9 10 11 ]
at
cada [ pd quem * 30 ]
mudecf 1
cada [ mudecl quem + 2 ]
cada [ mudect quem + 2 ]
raio
dt
fim
```



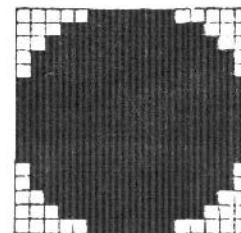
## TRABALHANDO COM SPRITES

Além das tartarugas o HOTLOGO possui dez figuras, chamadas **SPRITES**, que poderão ser utilizadas por você.

Apesar delas aceitarem todos os comandos do LOGO, a finalidade destas figuras não é substituir as tartarugas, mas sim oferecer recursos para animação na tela.

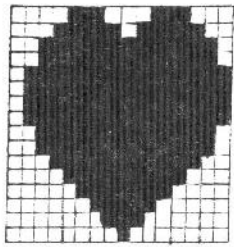
Para você ter acesso a um deles é preciso usar o comando **mundefig** seguido de um número correspondente ao relacionado na lista a seguir:

0



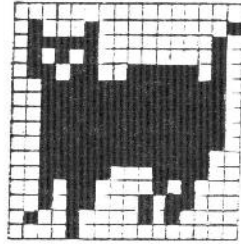
Círculo

1



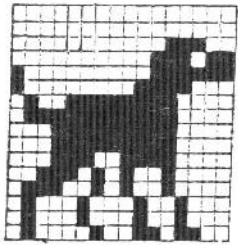
Coração

Gato



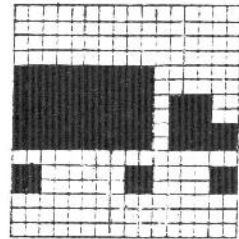
2

3



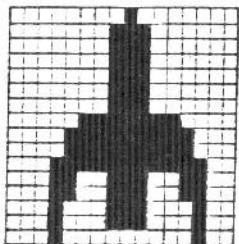
Cachorro

Caminhão



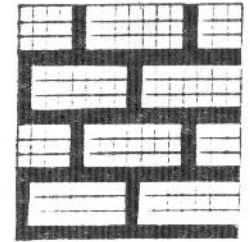
4

5



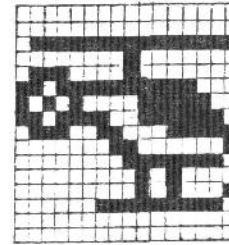
Foguete

Tijolos



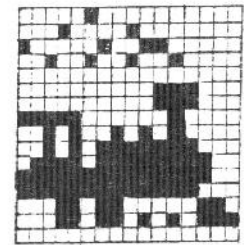
6

7



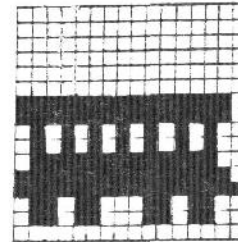
Helicóptero

Locomotiva



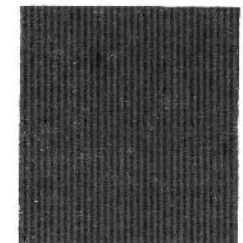
8

9



Vagão

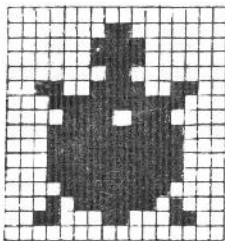
Quadrado cheio



10  
:  
35

36

59



Tartarugas com várias orientações

por exemplo, mudefig 2.

A posição dos sprites é fixa, isto é, o seu gatinho não gira como as tartarugas, embora ele risque em todas as direções.

Ficou confuso? Experimente pedir

```
pd 45
pf 50
```

e tenho certeza que você entenderá.

Com alguns dos comandos vistos no capítulo anterior podemos fazer com que esses sprites se movimentem.

Você poderá ver um caminhão andando pela tela se digitar:

```
mudefig 4
pd 90
mudevel 5
```

para que ele não risque, use un.

Mas não é só isso! No HOTLOGO é possível criar mais 26 sprites, numerados de 10 a 35.

Acompanhe comigo para ver como isso se dá, não esquecendo de limpar a tela através do comando rg.

Digite, então:

```
dt
edfig 10
```



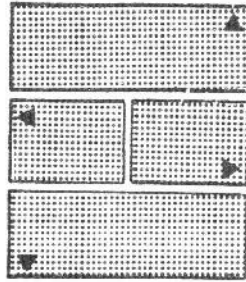
Pressione CTRL - K para transformar este quadrado, que é seu Editor de Figuras, numa grade de 16 por 16.

Repare que há um cursor piscando no canto esquerdo da tela.

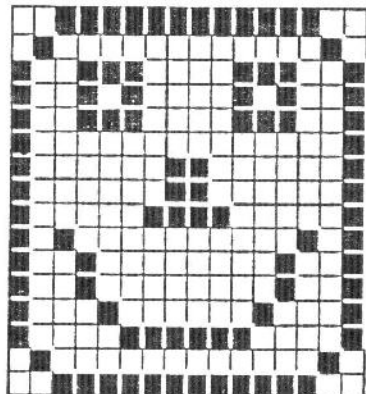
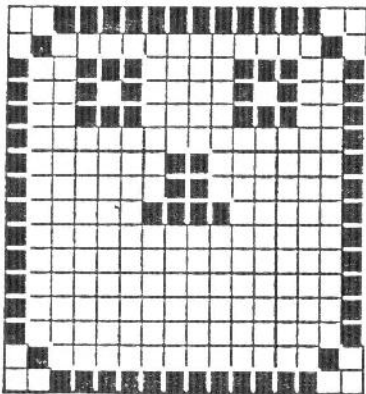
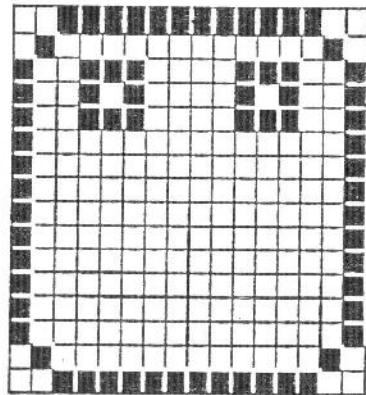
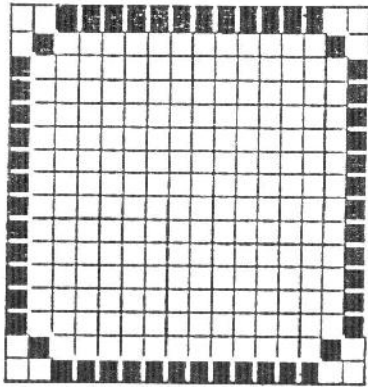
A cada vez que você pressionar a barra de espaço o quadradinho que está sob o cursor desaparecerá, começando, desta maneira, a construção de sua figura.

Para você movimentar o cursor dentro da grade, utilize as 4 teclas apropriadas.





Observe esta sequência e veja o resultado:



Se você gostou, aperte a tecla ESC para que esta figura fique guardada e pronta para ser usada a qualquer momento em que pedir mudefig 10.

Deste modo você saiu do Editor de Figuras e, para que ela apareça, utilize o comando at.

Se você não gostou, aperte as teclas CTRL e STOP para que esta figura não seja editada.

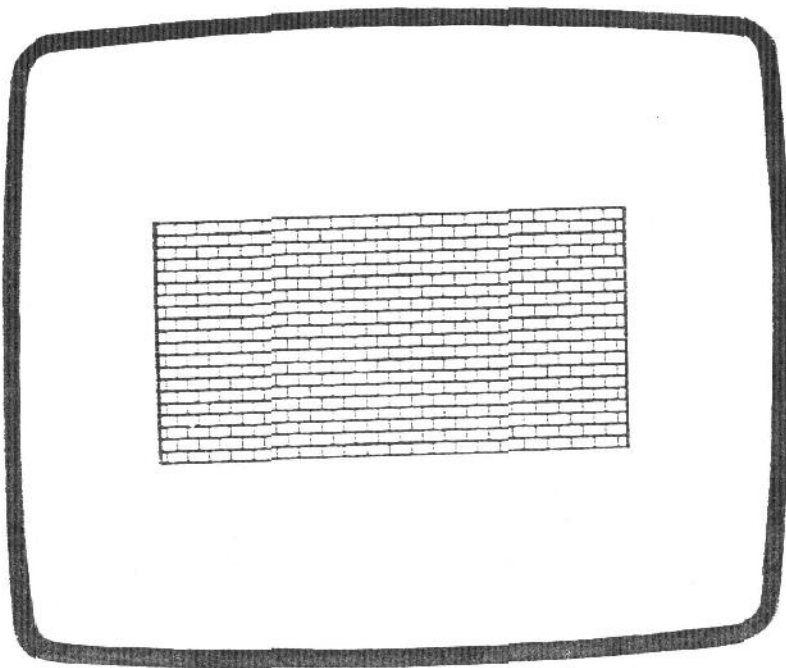
Fica por sua conta criar mais alguns sprites.

Um recurso interessante que o HOTLOGO possui é o comando carimbe, que deixa impressa na tela uma cópia do sprite com que se está trabalhando. Certifique-se se o comando funcionou fazendo a tartaruga ou o sprite desaparecer.

Para preencher toda uma área da tela ou uma figura qualquer, você deve usar carimbetudo.

Uma coisa importante: este comando funciona do mesmo modo que o **pinte**, ou seja, a tartaruga não pode estar sobre seu traço quando for preencher a figura.

Ex.: retângulo 40 70  
mudefig 6  
pd 45  
un  
pf 5  
ul  
carimbetudo



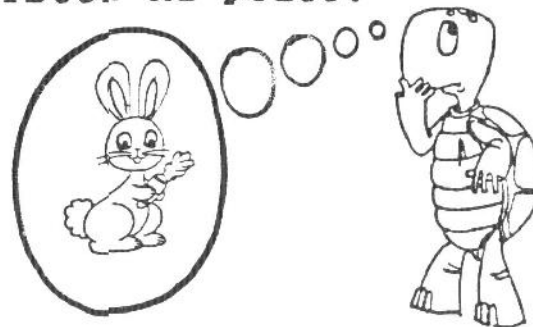
## SOM E ANIMAÇÃO



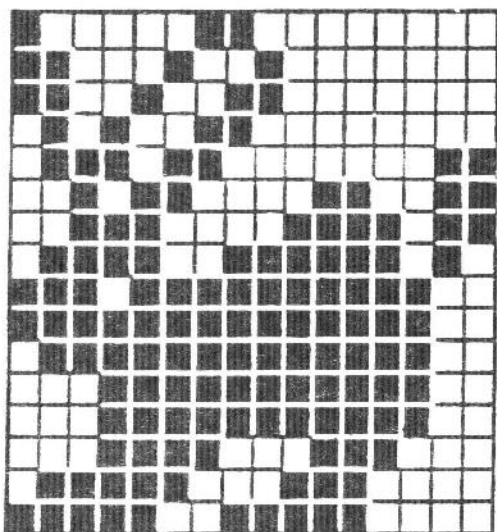
No capítulo anterior você viu como fazer para criar novos sprites. Através deste recurso vamos fazer um desenho animado.

Como fazer isto?

É preciso criar uma "figura tema" e colocá-la em diferentes posições, usando quantos sprites forem necessários para completar o movimento desejado. Por exemplo: você pode desenhar um coelho, utilizando os comandos vistos há pouco.

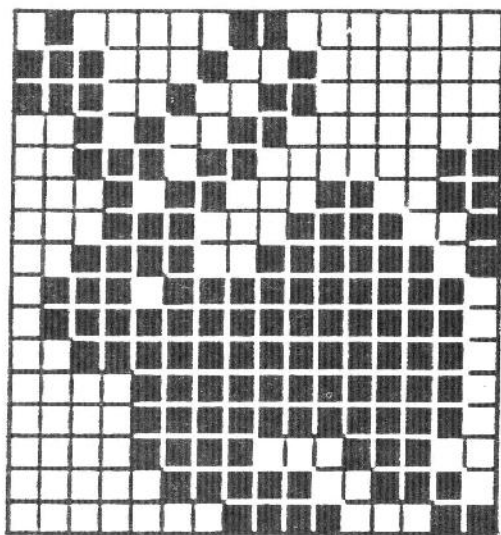


edfig 10



Em seguida chame novamente o Editor de Figuras:

edfig 11



Isto. Agora é só organizar o procedimento:

```
ap coelho
mudefig 10
espere 20
mudefig 11
espere 20
coelho
fim
```

Veja como ficou. Gostou?

Que tal agora dar som à figura? É só acrescentar estas instruções ao seu procedimento:

```
ap coelho
mudefig 10
toque 0 440 15 15
espere 20
mudefig 11
toque 0 880 15 15
espere 20
coelho
fim
```

E então? Não ficou mais interessante? Ah! Se você não ouviu nenhum som, aumente o volume de sua TV.

Vamos melhorar ainda mais nosso procedimento, fazendo o coelho correr.

Para isto precisamos criar outro procedimento - correndo - inserindo coelho como subprocedimento:

```
ap correndo
pe 90
un mudevel 8
coelho
fim
```

Agora peça correndo e veja o resultado.

Não é um barato?

Tente criar outros sprites dando-lhes som e animação. Você verá como é divertido.

Bom trabalho!!!

## ÚLTIMAS CONSIDERAÇÕES

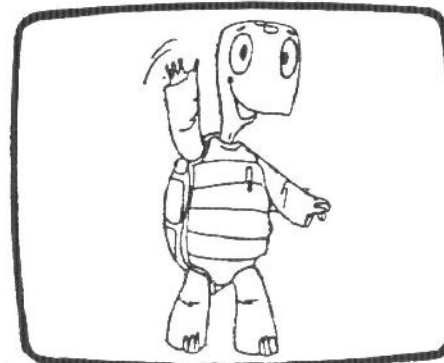


Se você olhar a lista de comandos relacionada no final deste livro verá que, por enquanto, não usou um grande número deles.

Programar em LOGO não quer dizer apenas definir procedimentos que façam desenhos. Sem dúvida é possível fazer coisas bem mais complexas. No entanto, como todo aprendizado é gradativo, foi preciso dividir a abordagem da linguagem em etapas.

Este livro corresponde à primeira delas.

Outras virão e, em pouco tempo, você poderá tornar-se um grande programador em LOGO.



Foi bom conhecer você.  
Nos veremos ainda  
no próximo livro.



## COMANDOS COMENTADOS

A seguir você encontrará em ordem alfabética um resumo dos comandos vistos neste livro e de todos os outros da parte gráfica do HOTLOGO.

**apaguedesenho** ou **ad**

- apaga somente o desenho da tela, deixando a tartaruga no lugar em que estava.

**apaguetexto** ou **att**

- apaga somente a tela de texto.

**apareçatat** ou **at**

- faz com que apareçam todas as tartarugas ativadas.

**aprenda** ou **ap nome**

- coloca o HOTLOGO no modo de definição sendo possível ensinar novos comandos à linguagem.

ex.: ap quadrado  
repita 4 [pf 50 pd 90]  
fim

**arquivos**

- lista todos os arquivos contidos num disco.

**atencãotat** ou **atat** *número ou  
[lista de números  
de 0 a 29].*

- ativa a(s) tartaruga(s) de  
número(s) indicado(s).

ex.: atat 2  
      atat [1 12 3]

**aumentelimit** ou **al**

- aumenta o limite da tela fazendo  
com que o vídeo seja apenas um  
pedaço, de uma tela maior.

ex.: al pf 700

**cada** *[lista de instruções]*

- pede às tartarugas ativadas que  
executem a lista de instruções.

**carimbe**

- carimba uma figura da tartaruga ou  
sprite ativado no lugar onde se  
encontra.

**carimbetudo**

- carimba num espaço limitado da tela  
a figura ativada, desde que seu lá-  
pis não esteja sobre o traço.

**carregue** "nome"

- passa para a memória do computador  
os arquivos gravados em disco.

**carreguec** "nome"

- passa para a memória do computador  
os arquivos gravados em fita.

**carreguedes** "nome"

- passa um desenho gravado em disco  
para a tela.

**congele**

- suspende a velocidade de todas as  
tartarugas ativadas até se digitar  
descongele.

**coorx**

- devolve a posição da tartaruga no  
eixo x (\*).

**coory**

- devolve a posição da tartaruga no  
eixo y (\*).

### cordebaixo

- devolve o número da cor existente sob o lápis da tartaruga (\*).

### cortat

- devolve o número da cor da tartaruga (\*).

### cordolápis ou cl

- devolve o número da cor do lápis que a tartaruga está usando (\*).

Obs.: (\*) se houver mais de uma tartaruga ativada no uso destes últimos cinco comandos, a resposta será referente à tartaruga de menor número.

### cordofundo ou cf

- devolve o número da cor do fundo da tela.

### desapareçatat ou dt

- faz desaparecer a imagem da(s) tartaruga(s) ativada(s).

### descongele

- faz com que as tartarugas voltem às suas respectivas velocidades interrompidas pelo comando congele.

### direção ou dç

- devolve, em graus, a direção da tartaruga ativada de menor número.

### direçãopara [x y]

- devolve, em graus, a direção a ser tomada pela tartaruga para atingir o ponto [x y].

### distância [x y]

- devolve a distância da tartaruga até o ponto [x y].

### edfig um número

- significa "edita figura". Coloca a figura de número pedido no modo de edição.

### edite ou ed [procedimento] ou [lista de procedimentos]

- edita um procedimento. Semelhante ao comando aprenda, só que admite listas.

### **espere número**

- faz acontecer uma pausa na execução das instruções de duração número 60 avos de segundo.

### **fig**

- devolve o número da figura ativada.

### **fim**

- utiliza-se para finalizar uma definição dentro de um procedimento.

### **gravedes "nome"**

- grava, num disco, o desenho realizado na tela.

### **gravetudo "nome"**

- grava, num disco, todos os procedimentos, variáveis e listas de propriedades que se encontrem na memória do computador.

### **gravetudoc "nome"**

- grava, numa fita, todos os procedimentos, variáveis e lista de propriedades que se encontrem na memória do computador.

### **mots**

- mostra o nome de todos os procedimentos contidos na memória do computador.

### **mudecf número**

- muda a cor do fundo da tela para a cor número desejada.

### **mudecl número**

- muda a cor do lápis da(s) tartaruga(s) ativada(s) para a cor número desejada.

### **mudect número**

- muda a cor da(s) tartaruga(s) ativada(s) para a cor número desejada.

### **mundefig número**

- muda a figura da(s) tartaruga(s) ativada(s) para a figura de número desejado.

### **mudepos [x y]**

- move a(s) tartaruga(s) ativada(s) para a posição [x y] determinada.

ex.: mudepos [30 57]



### **mudeproporção número**

- modifica para o número pedido o tamanho do passo da tartaruga no eixo x em relação ao eixo y. O valor habitual de "proporção" é 1,1.

### **mudedc número**

- leva a(s) tartaruga(s) ativada(s) para a direção pedida em graus.  
ex.: mudedc, 270

### **mudevel número**

- muda a velocidade da(s) tartaruga(s) ativada(s) conforme o número pedido. Este número pode variar entre -128 e 128.

### **mudex número**

- leva a(s) tartaruga(s) ativada(s) para a coordenada x pedida, mantendo sua posição no eixo y.

ex.: mudex 50 ou mudex -34

### **mudey número**

- leva a(s) tartaruga(s) ativada(s) para a coordenada y pedida, mantendo sua posição no eixo x.

ex.: mudey 80 ou mudey -57

### **paracentro ou pc**

- leva a(s) tartaruga(s) ativada(s) para o centro da tela (posição [0 0]).

### **paracentroa ou pc**

- leva a(s) tartaruga(s) ativada(s) para o centro da tela, apagando o desenho.

### **paradireita ou pd número**

- gira a(s) tartaruga(s) ativada(s) para a direita quantos graus pedirmos.

### **paraesquerda ou pe número**

- gira a(s) tartaruga(s) ativada(s) para a esquerda quantos graus pedirmos.

### **parafrente ou pf número**

- faz a(s) tartaruga(s) ativada(s) se locomover(em) para frente tantos passos quantos pedirmos.

### **paratrás ou pt número**

- faz a(s) tartaruga(s) ativada(s) se locomover(em) para trás tantos passos quantos pedirmos.

## **pinte**

- preenche um espaço limitado da tela com a cor do lápis da tartaruga, desde que ela não esteja sobre seu traço.

## **ponhaponto ou pp [x y]**

- coloca um ponto na coordenada [x y]

## **posição**

- devolve a posição da tartaruga ativada de menor número.

## **proporção**

- devolve a atual proporção entre o passo vertical e o horizontal da tartaruga.

## **quem**

- devolve a lista de tartarugas ativadas no momento.

## **repita número [lista de instruções]**

- repete um número de vezes uma determinada lista de instruções.

## **rg**

- apaga o desenho da tela, devolve a tartaruga ao centro da mesma, restaura a cor do lápis, cor do fundo, cor da tartaruga, enfim, restaura o sistema de gráfico.

## **tartaruga ou tat**

- apaga as telas de texto e gráfica devolvendo a tartaruga à sua posição inicial.

## **tirelimite ou tl**

- limita a área de trabalho da tela gráfica ao tamanho real do vídeo.

## **toque canal frequência volume tempo**

- emite um som pelo canal pedido, na frequência, volume e tempo determinados.

## **useborracha ou ub**

- apaga o rastro deixado pela tartaruga ao passar novamente por cima do mesmo.

**uselápis** ou **ul**

- volta a riscar a tela usando a mesma cor anteriormente especificada.

**useinversor** ou **ui**

- faz com que a tartaruga apague o rastro deixado ao passar sobre ele ou, trace novas linhas.

**usenada** ou **un**

- permite que a tartaruga se desloque na tela sem traçar linhas.

**vel**

- indica a velocidade atual da tartaruga.

**velx**

- indica a velocidade da tartaruga em relação à linha horizontal.

**vely**

- indica a velocidade da tartaruga em relação à linha vertical.

**évisível**

- responde "verd" sempre que a tartaruga estiver visível; caso contrário responderá falso.

## LISTA DE PRIMITIVAS

### ■ COMANDOS DA PARTE GRÁFICA

**apaguedesenho** ou **ad**

**apareçatat** ou **at**

**atençãotat** ou **atat**

**aumentelimit** ou **al**

**cada**

**carimbe**

**carimbetudo**

**congele**

**coorx**

**coory**

**copiafig**

**cordebaixo**

**cortat**

**cordolápis** ou **cl**

**cordofundo** ou **cf**

**criafigl**

**desapareçatat** ou **dt**

**descongele**

direção ou dc

direção para [x y]

distância [x y]

edfig número

em.colisão [dois números]

fig

inverta

listafig

mudecf número

mudecl número

mudect número

mudefig número

mudepos [x y]

mudeproporção número

mudedç número

mudevel número

mudex número

mudey número

paracentro ou pc

paracentroa ou pca

paradireita, pd

paraesquerda, pe

parafrente, pf

paratrás, pt

pinte

ponhaponto ou pp [x y]

posição

proporção

quando

quem

rg

tartaruga, tat

tirelimite, tl

todas

useborracha, ub

uselápis, ul

useinversor, ui

usenada, un

vel

velx

vely

évisível

■ LISTAS E PALAVRAS

ascii *carácter*

caractere ou car *número*

elemento ou elem *número objeto*

élista

énúmero

épalavra

évazia

junteno fim ou jf *objeto lista*

junteno início ou ji *objeto lista*

lista *objeto1 objeto2*

num.elem ou nel *objeto*

palavra ou pal *palavra1 palavra2*

primeiro ou pri *objeto*

são iguais *objeto1 objeto2*

semprimeiro ou sp *objeto*

semúltimo ou su *objeto*

sentença ou sn *objeto1 objeto2*

último ou ult *objeto*

● MATEMÁTICA

arctan *número*

arredonde *número*

cos *número*

diferença *número1 número2*

int *número*

produto *número1 número2*

quociente *número1 número2*

raizq ou rq *número*

reproduza

resto *número 1 número2*

sen *número*

soma *número1 número2*

sorteieaté *número*

troquesinal *número ou nome*

● VARIÁVEIS

atribua ou atr

conteúdo

coloque

temvalor

● OPERAÇÕES LÓGICAS

*algum pred1 pred2*

*não pred*

*e pred1 pred2*

● ESPAÇO DE TRABALHO

*eln nome ou [lista de nomes]*

*elns*

*elps*

*eltudo*

*liberemem*

*memlivre*

*mons*

*mop nome*

*mops*

*mots*

*motudo*

*primitivas*

● DEFINIÇÃO E EDIÇÃO

**aprenda** ou **ap** "nome"

**defina** nome lista

**edite** ou **ed** nome ou [lista de nomes]

**edns**

**fim**

**éprimitiva**

**éprocedimento**

**texto**

● CONDICIONAIS E CONTROLE DE EXECUÇÃO

**care**

**envie**

**faça** lista

**falso**

**espere** número

**nível** inicial

**repita** número  
[lista de instruções]

**se**

**verd**



● IMPRESSORA, TECLADO  
TELA E ARQUIVOS

apaguetexto ou att

arquivos

carregue "nome

carreguec "nome

carreguedes "nome

comimpressora

cursor

eliminarq nome

escreva ou esc objeto

gravedes "nome

gravetudo "nome

gravetudoc "nome

line

mostra ou mo objeto

mostrearq nome

mudecursor [coluna linha]

mudeteto linha

ponha objeto

semimpressora

temcar

versão

● LISTA DE PROPRIEDADES

coloqueprop *nome prop objeto*

moprop

propriedade ou prop *nome prop*

retireprop ou rp *nome prop*

retiretodasprop ou rtp

todasprop *nome*

● MÚSICA

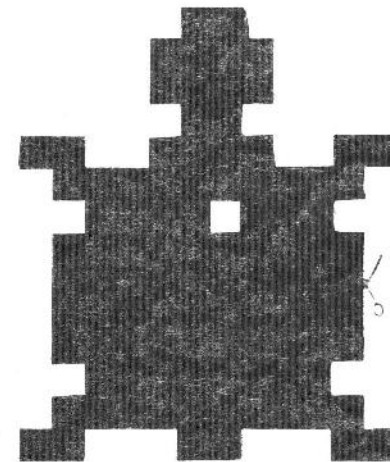
som *tom volume tempo*

toque *canal frequência volume tempo*

## ATIVIDADE 1

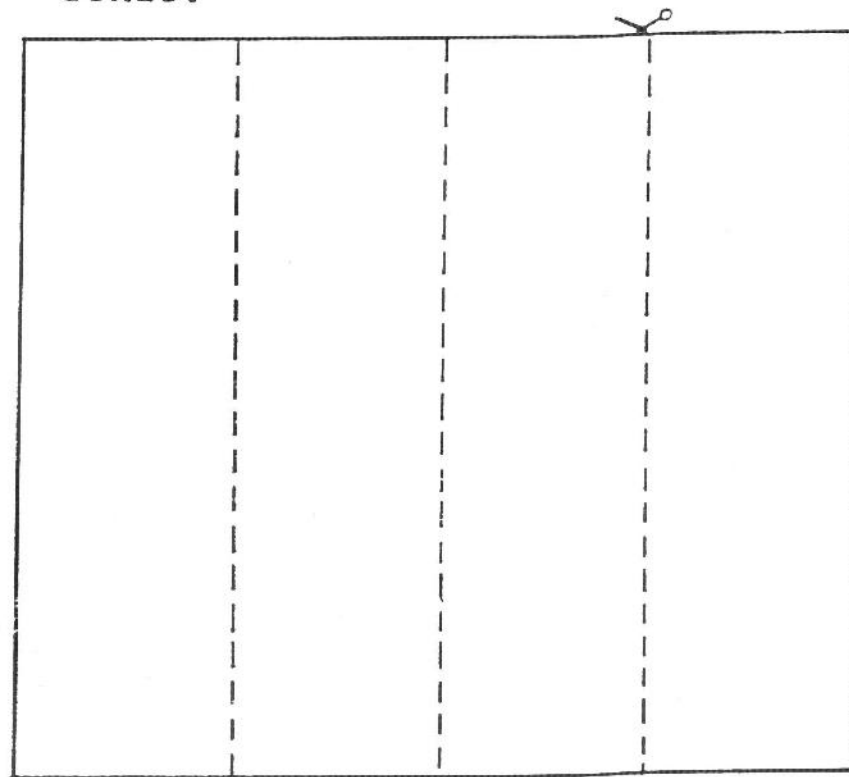
Recorte esta tartaruga e cole na parte de cima de um lápis.

Use seu lápis-tartaruga em todas as atividades ligadas ao LOGO.

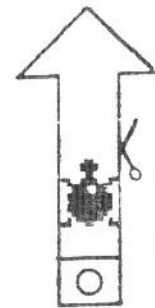
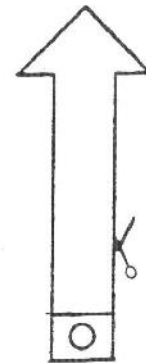
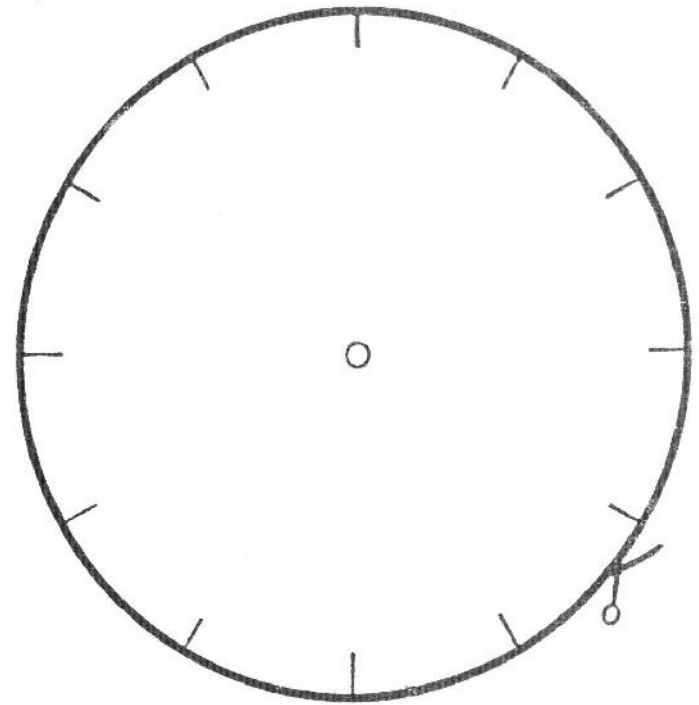


## ATIVIDADE 2

- 1) Recorte o retângulo abaixo;
- 2) Pegue um dos lados e dobre até o meio;
- 3) Faça o mesmo com o outro lado;
- 4) Mantenha o papel assim e, usando um lápis e uma régua, faça um traço cruzando as partes dobradas e marcando o início com um X e o seu final com uma tartaruga;
- 5) Desdobre o papel e veja o efeito final.



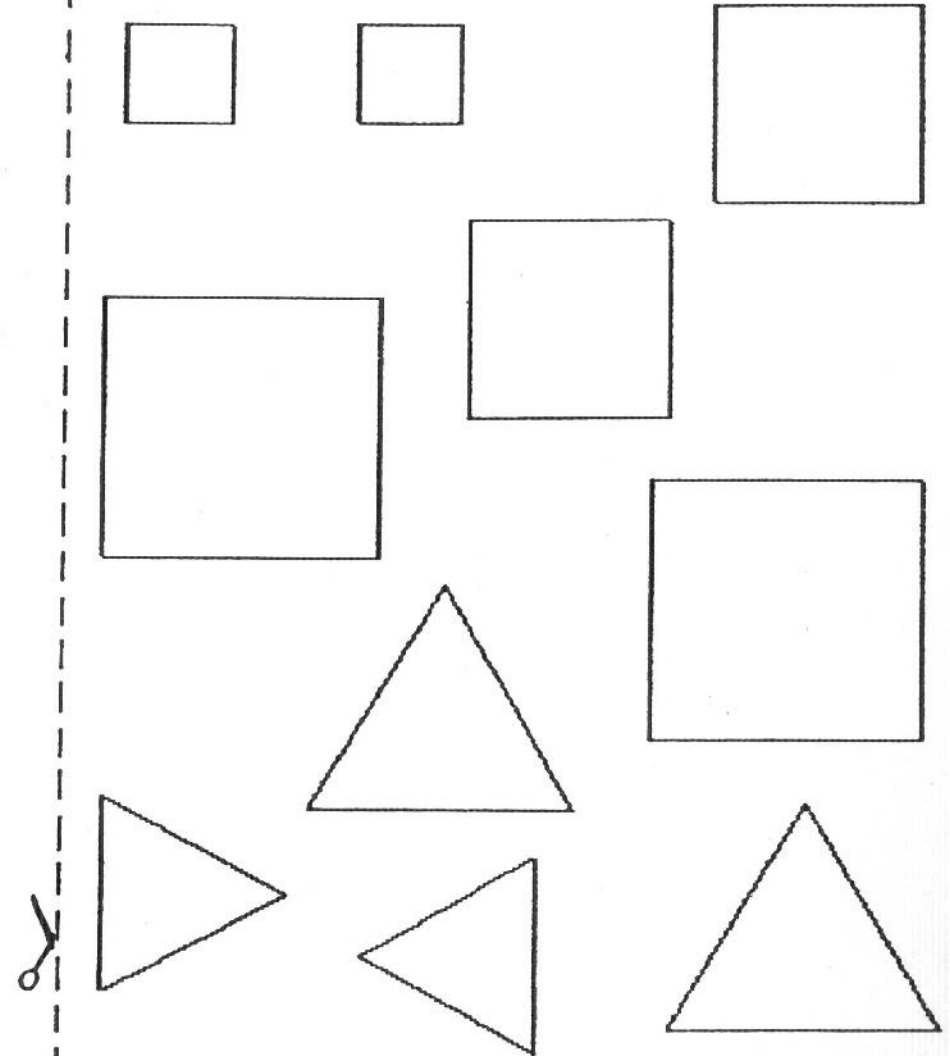
ATIVIDADE 3

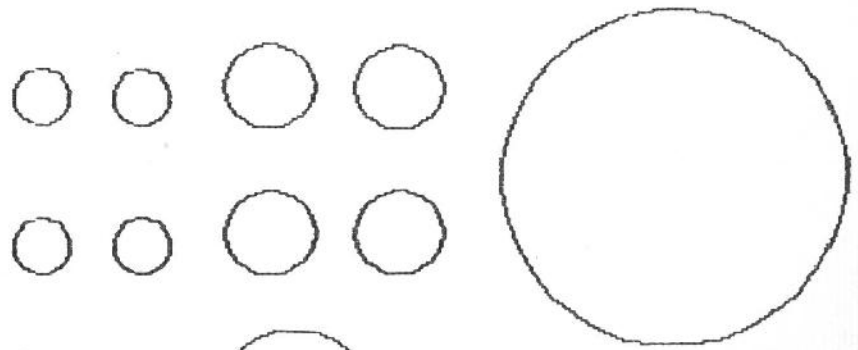
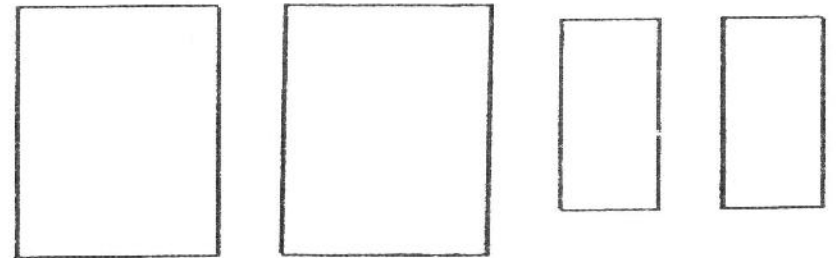
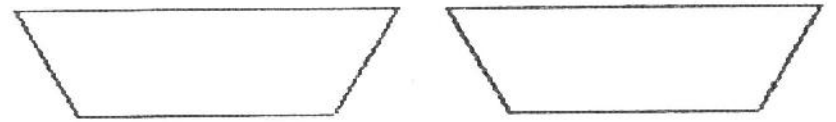
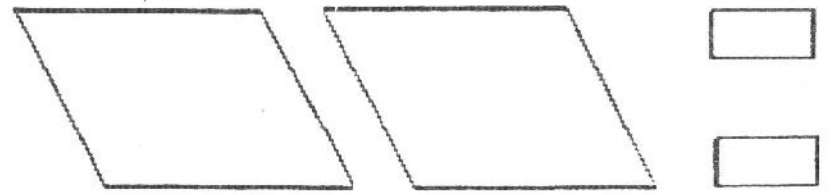


## ATIVIDADE 4

Recorte estas figuras e, para que elas fiquem mais bonitas, pinte com suas cores preferidas.

Brinque com elas e veja quantos desenhos diferentes você é capaz de montar como um quebra-cabeça.





## OUTROS LIVROS DA COLEÇÃO MSX

### USANDO DISK DRIVE NO MSX

Os três sistemas operacionais disponíveis para a linha MSX, comentados detalhadamente de forma fácil e didática, desde a instalação até os usos mais sofisticados. E um capítulo com um completo dicionário de todos os comandos do DSK-BASIC, HB-DOS e do HB-MCP.

### APROFUNDANDO-SE NO MSX

Livro dedicado ao leitor que já conhece BASIC e quer saber todos os detalhes da máquina. Além de uma grande quantidade de dicas e informações explicadas de forma completa e didática.

### COLEÇÃO DE PROGRAMAS VOL. I

Uma coletânea de programas dedicada aos usuários principiantes do MSX. Jogos, músicas, desenhos, apresentados de maneira simples e didática, com instruções de digitação e análise dos programas.

### COLEÇÃO DE PROGRAMAS VOL. II

Um conjunto completo de programas, com rotinas em Basic e em Linguagem de Máquina. Jogos, programas didáticos e profissionais, comentados detalhadamente.



## JOGOS DE HABILIDADE PARA MSX

Coletânea de jogos de ação-inteligência de nível profissional, usando sempre que necessário os recursos da Linguagem de Máquina.

### TABELA DE MNEMÔNICOS Z80

Uma tabela com todos os códigos mnemônicos do ASSEMBLY Z80. Imprescindível para quem está começando a programar em linguagem de máquina e não dispõe de um programa assembler.

### HOTWORD - Processador de textos

Uma completa explanação sobre o Processador de Textos HOTWORD, suas funções e utilidades. Após a leitura deste livro, o usuário estará em condições de usar com total eficácia todos os recursos do software.

### HOTPLAN - Planilha de Cálculos

Livro que descreve passo a passo os detalhes de instalação e utilização do HOTPLAN, a planilha de cálculos para o HOTBIT.

### HOTDATA - Banco de Dados

Um livro sobre o gerenciador de banco de dados desenvolvido para o HOTBIT e compatíveis MSX. Desde a instalação do programa até aplicações sofisticadas com exemplos e dicas.

## COMO RECEBER NOSSO BOLETIM

Se você quiser receber sistematicamente nosso boletim informativo com dicas de programação e novidades para os micros MSX, envie-nos o cupom abaixo com seu nome e endereço.

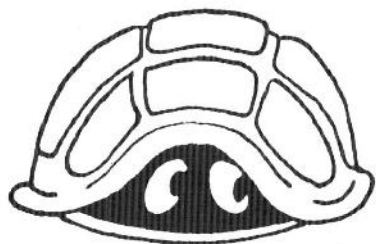
Nome: .....  
End.: .....  
Bairro: ..... CEP: .....  
Cidade: ..... Estado:.....

Se você é professor e se interessa por informática aplicada à educação, preencha os dados abaixo e remeta-o para nós junto com o cupom.

Escola(s) em que leciona:  
.....  
.....  
.....  
Matéria(s) que leciona:  
.....  
.....  
Séries para as quais leciona:  
.....  
.....  
Áreas de interesse:  
.....  
.....  
.....

ALEPH Publicações e  
Assessoria Pedagógica Ltda.  
Caixa Postal: 20707 - CEP: 01498  
São Paulo - S.P.

Tel.: (011) - 813-4555 / 212-4917



Este livro foi impresso pelo processo  
OFF-SET utilizando-se de filmes pro-  
duzidos na copiadora SHARPMASTER mo-  
dêlo SF 756.

## COLEÇÃO MSX

### HOT LOGO

GODOY- LACERDA - LEPÍSCOPO - MENDES

Desenvolvido por Seymour Papert, do M.I.T., especificamente para aplicações didáticas e pedagógicas, o LOGO transformou-se atualmente numa das mais completas e poderosas linguagens de programação disponível para microcomputadores.

As implementações que o LOGO sofreu desde sua criação, entretanto, não eliminaram suas principais características pedagógicas e é justamente a elas que se deve a grande aceitação que essa linguagem tem tido, não apenas pelas crianças, mas também por adultos.

Este livro, elaborado por uma competente equipe de pedagogos, apresenta de forma clara e suave, através de um constante diálogo com a criança, os aspectos essenciais do HOT LOGO, uma das melhores e mais completas versões do LOGO, desenvolvida especialmente para o HOT BIT e compatíveis MSX.

Com o cuidado peculiar a quem tem por hábito ser paciente e metucioso, os autores conseguem introduzir seus leitores infantis, seus pais e seus mestres no fantástico universo da linguagem LOGO.