

UNIVERSIDADE DE SÃO PAULO

Noções Básicas de Programação LOGO
versão 1.0

SOLANGE REZENDE RODRIGUES

Nº 11

NOTAS DIDÁTICAS



Instituto de Ciências Matemáticas de São Carlos



Instituto de Ciências Matemáticas de São Carlos

ISSN - 0103-2585

Noções Básicas de Programação LOGO
versão 1.0

SOLANGE REZENDE RODRIGUES

Nº 11

NOTAS DIDÁTICAS DO ICMSC

São Carlos
JUN. / 1993

Noções Básicas de Programação LOGO

Solange Rezende Rodrigues

Universidade de São Paulo / ILTC

Instituto de Ciências Matemáticas de São Carlos

Departamento de Ciências de Computação e Estatística

Versão 1.0

São Carlos – Maio 1993

Esta Nota Didática é dedicada, com carinho especial:
ao *André Sarkis Müller*
à *Lia Martucci de Amorim* e
à *Sandra Brockon*.
pelo empenho e dedicação em aprender LOGO.

Conteúdo

1	Introdução	1
2	Como Carregar e Como Sair do LOGOWRITER	1
3	Comandos Básicos ou Primitivas	2
3.1	Campo de Movimento da Tartaruga	3
3.2	Lápis	3
3.3	Voltar para Casa	3
3.4	Apagar o Risco do Lápis	3
4	Construção de Programas	4
4.1	Como Construir um Programa	4
4.2	Projetos Estruturados em Sub-programas	5
5	Procedimentos Aritméticos	6
6	Variáveis	7
7	Recursos Utilizados para Animação	8
7.1	Som	8
7.2	Múltiplas Tartarugas	10
7.3	Formas	10
7.4	Animação	11
8	Recursão	12
9	Gravação de Programas	13
10	Impressão	13
11	Exemplos de Programas em LOGO	14

12 Exercícios	24
13 Conclusões	25
Referências	25

1 Introdução

A linguagem de programação LOGO foi especialmente concebida para ser utilizada por crianças, a partir dos 5 anos, permitindo um contato com o computador. Nesta linguagem, o papel central é desempenhado pela "tartaruga", um pequeno ser a quem são transmitidos os nossos comandos e instruções. Por exemplo, para fazer a tartaruga deslocar-se para frente escrevemos PF 40, o que significa "Para frente 40 passos". A tartaruga é um ser interativo que pode ser manipulado pelo aprendiz, fornecendo um ambiente para uma aprendizagem ativa [Papert 80].

Nesta Nota Didática são abordadas apenas as noções básicas da linguagem LOGO. Os aspectos educativos da linguagem foge ao escopo deste trabalho. Apenas é dada uma breve explicação de alguns conceitos e os comandos mais simples da linguagem — o LOGOWRITER.

A seguir é descrita a organização do trabalho. A seção 2 ensina como carregar o LOGO na memória e como sair do sistema no final da seção de trabalho. A seção 3 mostra algumas primitivas básicas do LOGO. A seção 4 ensina como contruir programas utilizando as primitivas básicas vistas na seção 3 e como estruturar os programas em sub-programas (programas menores). A seção 5 mostra o uso de operadores aritméticos. A seção 6 ensina como usar variáveis para uma melhor utilização das técnicas vistas nas seções anteriores. Na seção 7 são mostrados alguns recursos de som e animação usando múltiplas tartarugas. A seção 8 é dedicada ao uso de recursão. A seção 9 mostra como gravar os programas criados e a seção 10 como imprimí-los. A seção 11 contém alguns exercícios. O objetivo destes exercícios é recordar os conceitos para verificar o aprendizado. Cabe, portanto, a cada aprendiz deixar a sua imaginação livre e tentar ver o que ele consegue criar com os comandos vistos em cada seção.

2 Como Carregar e Como Sair do LOGOWRITER

Para iniciar o trabalho com o LOGO digite `logowr` que carrega o LOGO para a memória. Em seguida o sistema — através do LOGO — pedirá para colocar o disco de trabalho no drive B.

Para iniciar uma seção de trabalho você pode selecionar uma página de trabalho que já existe (com auxílio das setas) ou selecionar uma nova (escolhendo a palavra NOVA do menu).

Para sair do LOGO, digite DOS. Se você selecionou a opção NOVA é necessário dar um nome à página, antes de digitar DOS. O comando

- np "nome_do_arquivo"

é utilizado para dar nome à página.

3 Comandos Básicos ou Primitivas

No LOGO a tartaruga "sabe" executar certas tarefas utilizando as funções primitivas. Por exemplo, ela sabe Andar para Frente, bastando para isso dar-lhe a ordem PF 50 <Enter>.

- PF n

é um comando de movimento que significa Andar Para Frente n passos.

O deslocamento da tartaruga é feito a partir da posição onde ela se encontra e na direção da sua cabeça. Quando iniciamos o trabalho em Logo, a tartaruga está "em casa": trata-se do centro da tela, que corresponde ao ponto (0,0) e orientação 0 graus. O estado da tartaruga é definido por dois elementos: a sua posição e a sua orientação.

A tartaruga conhece outros comandos de movimento como:

- PT n (Anda para Trás n passos)
- VD m (Vire à Direita m graus)
- VE m (Vire à Esquerda m graus)

Após algum tempo de trabalho, a tela (chamada no LOGO de "ecrã") fica com um aspecto um pouco confuso. Para limpá-la utilize o comando:

- LE (Limpa a Ecrã — tela)

Como primitiva existe também o comando de repetição, que economiza muito trabalho em diversas ocasiões, como por exemplo:

- REPETE n [.....]
Ex: REPETE 4 [PF 50 VD 90]

3.1 Campo de Movimento da Tartaruga

A tartaruga está limitada a uma região de trabalho retangular ligeiramente inferior às dimensões da ecrã. Verifique o que acontece quando você executa os seguintes comandos:

```
LE  
VD 10  
PF 600
```

3.2 Lápis

Quando a tartaruga se desloca deixa um traço na ecrã. É como se tivesse um lápis que vai riscando o chão por onde se desloca. Mas a tartaruga pode levantar o lápis.

- LC (Lápis em Cima)
- LB (Lápis em Baixo)

3.3 Voltar para Casa

A tartaruga tem sempre um meio de voltar rapidamente para casa, no meio da tela (ponto de coordenadas (0,0)):

- EC (Em Casa)

Além disso a tartaruga poderá ser ou não visível, usando os comandos:

- ET (Esconde Tartaruga)
- MT (Mostra Tartaruga)

3.4 Apagar o Risco do Lápis

A tartaruga tem a possibilidade de apagar traços já existentes. É como se a tartaruga tivesse uma borracha na ponta.

- LA (Lápis para Apagar)

4 Construção de Programas

4.1 Como Construir um Programa

Como visto anteriormente, a tartaruga consegue realizar várias tarefas com base nos conhecimentos das primitivas ou comandos, (pf, vd, le, etc...). Além destes comandos podemos ensinar-lhe novos programas com base nas primitivas já conhecidas. Estes novos programas poderão por sua vez ser usados para construir outros programas mais complexos, e assim sucessivamente.

No LOGOWRITER a construção de programas é realizada na página de programação **O Outro Lado**. O comando que permite ir para a página de programação é:

- <CTRL> <F> simultaneamente

Com este mesmo comando é possível voltar a página de gráficos e texto.

Exemplo:

Construir um programa para desenhar um DEGRAU:

- Primeiro passo: <CTRL> <F> para passar à página de programação;
- Segundo passo: escrever o programa. Todos os programas começam por TO <nome do programa> e terminam com FIM;

```
TO DEGRAU
  PF 50
  VD 90
  PF 50
FIM
```

- Terceiro passo: <CTRL> <F> para retornar à página gráfica;
- Quarto passo: Digitar DEGRAU.

Obs:

1. Na página de programação podemos escrever comentários sobre os programas.
2. Para limpar a área de comandos (isto é, as três últimas linhas da ecrã) temos o comando AC (Apagar Comandos).

4.2 Projetos Estruturados em Sub-programas

Podemos construir programas utilizando programas menores. Esta maneira de resolver problemas permite decompor um problema em sub-problemas mais simples.

Exemplo:

Desenvolver um projeto, utilizando os comandos ou programas primitivos do Logo para desenhar um bairro (um conjunto de casas).

TO QUADRADO

REPETE 4 [PF 30 VD 90]

FIM

TO TELHADO

VD 30

REPETE 3 [PF 30 VD 120]

FIM

TO PORTA

PF 10

VD 90

PF 5

VD 90

PF 10

FIM

TO CASINHA

LE

QUADRADO

PF 30

TELHADO

VD 60

LC

PF 10

VE 90

PT 30

LB

PORTA

ET

FIM

```
TO CASINHAS
  REPETE 3 [ CASINHA VE 90 PF 20 VE 90 ]
FIM
```

```
TO BAIRRO
  CASINHAS
  VE 90
  PF 15
  VD 90
  LC
  PF 10
  LB
  CASINHAS
FIM
```

5 Procedimentos Aritméticos

A linguagem LOGO permite a execução das operações aritméticas usuais. Podemos colocar o computador a fazer contas ou até a funcionar como uma calculadora. Por exemplo, para adicionar dois números e saber o respectivo resultado, pode-se fazer:

```
410 + 234 <Enter>
```

Mas o computador responde com a mensagem: "Não dizes o que queres fazer com 644". Nesse caso o computador executou a operação de adição e ficou com o resultado sem saber o que fazer. Se for escrito `ES 410 + 234 <Enter>`, o computador escreverá na tela o valor 644.

É sempre conveniente deixar um espaço entre o comando `ES`, os valores numéricos, e os sinais das operações que são representadas por:

1. + adição
2. - subtração
3. * multiplicação
4. / divisão

Várias operações podem ser combinadas e executadas de uma só vez. Por exemplo:

```
ES 321 + 3 / 21 * 5 - 10 <Enter>
```

Deve-se tomar cuidado com a ordem na qual as operações são executadas, que obedece uma regra previamente definida:

1. Primeiro as divisões
2. Depois as multiplicações
3. Depois as subtrações
4. Finalmente as adições

Esta ordem pode ser modificada utilizando parênteses. Por exemplo, pode-se querer que no exemplo anterior a multiplicação seja executada em primeiro lugar. Nesse caso devem ser utilizados parênteses:

ES 321 + 3 / (21 * 5) 10 <Enter>

Em LOGO é possível trabalhar com números naturais, inteiros ou decimais.

O comando ES (Escreve) também pode ser utilizado para escrever palavras ou frases.

- ES "palavra <Enter> ou
- ES [Uma frase] <Enter>

Para limpar o texto da tela basta utilizar o comando:

- LT Limpa Texto, mas não apaga os gráficos

Para apagar frases da área de comandos pode-se utilizar o comando:

- AC (Apagar Comandos)

6 Variáveis

Até agora, todos os programas que foram construídos são aplicados a uma única situação. Por exemplo, o programa QUADRADO permite desenhar um quadrado de lado 50:

```
TO QUADRADO
  REPETE 4 [ PF 50 VD 90 ]
FIM
```

Para aumentar a potência dos programas construídos em LOGO deve-se aumentar a sua capacidade de ser aplicado a um grande número de situações, para solucionar este problema é necessário introduzir variáveis. Por exemplo, é mais interessante ter um programa que desenha quadrados de diversos tamanhos (dimensões), do que ter um programa que desenha apenas quadrado de lado 20.

```
TO QUAD20
  REPETE 4 [ PF 20 VD 90 ]
FIM
```

Para construir um programa que desenha quadrados de qualquer tamanho é necessário introduzir uma variável, que pode ser :LADO. O novo programa é o QUAD.

```
TO QUAD :LADO
  REPETE 4 [ PF :LADO VD 90 ]
FIM
```

Observe que:

1. Ao dar o nome ao programa é necessário declarar as variáveis que serão utilizadas;
TO QUAD :LADO
2. O nome das variáveis deve ser antecedido por dois pontos (:). Uma variável que chamamos por LARG deve ser introduzida no programa como :LARG. Se escrevemos apenas LARG, o computador assumirá que se trata do nome de um programa e não de uma variável.

Para rodar um programa com variáveis basta escrever o nome do programa seguido do valor que queremos atribuir à variável. Por exemplo, para desenhar um quadrado de lado 20 basta escrever QUAD 20 na área de comandos.

Pode-se construir programas com mais de uma variável, basta especificá-las.

7 Recursos Utilizados para Animação

7.1 Som

O computador pode ser programado para emitir sons. O comando usado é SOM que exige dois valores: o primeiro é a frequência do som e o segundo é o tempo

de duração. A frequência do som define a natureza mais grave ou mais aguda do som a emitir, e é dada por um número entre 0 e 9999. A duração tem por base o valor 20, que corresponde a mais ou menos 1 segundo.

Por exemplo, para emitir um som musical LA durante 1 segundo pode-se escrever:

SOM 440 20

Alguns valores de frequência das notas musicais encontram-se na tabela abaixo:

Nota	Frequências		
LA	220	440	880
LA	233	446	932
SI	246	439	987
DO	261	523	1046
DO	227	554	1108
RE	293	587	1174
RE	311	622	1244
MI	329	659	1318
FA	349	698	1397
FA	370	740	1480
SOL	392	784	1568
SOL	415	830	1661

Por exemplo, o programa AMB simula o som emitido por uma ambulância:

TO AMB

REPETE 10 [SOM 220 15 SOM 440 15]

FIM

Outro comando interessante no trabalho com sons é o comando ESPERA. Este comando exige um valor numérico, que corresponde ao tempo de duração da pausa permitida.

Por exemplo, para o computador fazer uma pausa de 1 segundo em um programa, o comando ESPERA 20 pode ser usado da seguinte maneira:

TO BARULHO

REPETE 10 [SOM 440 5 SOM 220 5 SOM 110 5 ESPERA 20]

FIM

7.2 Múltiplas Tartarugas

Com o LOGOWRITER, pode-se trabalhar com até quatro tartarugas ao mesmo tempo. As tartarugas têm um número, 0, 1, 2, ou 3. A tartaruga 0 é aquela que já aparece quando se entra no LOGOWRITER. Para que a segunda apareça, é só fazer:

- TA 1 MT

Para todas as tartarugas aparecerem: TA [1 2 3] MT

As tartarugas podem fazer todas o mesmo desenho, juntas. Um exemplo é o programa BAILE.

```
TO BAILE
  TA [ 0 1 2 3 ] MT
  LC PF 50
  REPETE 10 [REPETE 5 [SOM 440 5 PF 20 SOM 220 5 VD 72
    SOM 110 5 ESPERA 10] ]
FIM
```

Experimente mudar o programa. Por exemplo:

```
TO NOVOBAILE
  TA [ 0 1 2 3 ] MT
  LC PF 50
  TA [ 1 3 ]
  REPETE 10 [REPETE 5 [SOM 440 5 PF 20 SOM 220 5 VD 72
    SOM 110 5 ESPERA 10] ]
FIM
```

7.3 Formas

Além da forma da tartaruga, pode-se ter diferentes formas do desenho desempenhando a função da tartaruga. Para isso, utiliza-se o comando FIXAFORMA. Esse comando exige um valor, correspondente ao número da nova forma do desenho. Por exemplo:

- TA 0 FIXAFORMA 22

fixa a forma de um coelho, para a tartaruga 0.

Para ver as formas de desenho existentes, utiliza-se o comando FORMAS. Para escolher uma forma, ou desenhar uma nova, utilize <CTRL> <F> para ver a página de programação das formas. As teclas PgUp e PgDn são usadas para mostrar as outras formas.

Novas formas podem ser desenhadas utilizando a barra de espaço e as teclas normais de movimentação de cursor. Terminado o desenho, tecele <Esc> para voltar à página de programação.

O mesmo procedimento pode ser utilizado para modificar as formas já existentes.

7.4 Animação

TO ESTRADA

TA 0 LC PF 20

VD 90 LB PF 400

VD 90 LC PF 30

VD 90 LB PF 400

VD 90 LC PF 15 VD 90

REPETE 50 [PF 5 LB PF 5 LC]

ET

FIM

TO CARROS

TA 1 FIXAFORMA 26

VD 90 LC PF 50 MT

TA 2 FIXAFORMA 27

VD 90 LC MT

TA [1 2]

REPETE 1000 [PF 2 SOM 40 2 ESPERA 2]

FIM

TO ESTRADACARRO

ESTRADA

CARROS

FIM

Digite esses programas e veja o que acontece. Aqui, pode-se também mudar a velocidade das tartarugas, fazendo com que cada uma ande em uma velocidade diferente, combinando o comando PF com a tartaruga escolhida. Por exemplo, modifique o programa CARROS:

TO CARROS

TA 1 FIXAFORMA 26

VD 90 LC PF 50 MT

TA 2 FIXAFORMA 27

VD 90 LC MT

REPETE 1000 [TA 1 PF 2 TA 2 PF 3 SOM 40 2 ESPERA 2]

FIM

e observe que os carros estão com velocidades diferentes.

8 Recursão

Suponhamos que alguém quer desenhar quatro quadrados na tela, que sejam cada vez maiores, começando no mesmo ponto. Pode-se fazer o seguinte:

TO DESENHA :L

QUADRADO :L

QUADRADO :L + 10

QUADRADO :L + 10 + 10

QUADRADO :L + 10 + 10 + 10

FIM

Como fazer isso sem chamar o programa QUADRADO tantas vezes? Usando recursão, que é o ato de repetir uma tarefa, "relembrar", ou fazer de novo algo que já foi feito anteriormente.

Assim, o programa desenha ficaria:

```
TO DESENHA :L
  QUADRADO :L
  DESENHA :L + 10
FIM
```

Observe que esse programa fica desenhando quadrados “eternamente”, a não ser que você interrompa com <CTRL> <Break>. O que fazer? Bem, uma solução é controlar o tamanho do quadrado, ou seja, a cada desenho, verificar se o quadrado já chegou no tamanho máximo permitido, utilizando o comando SE. Então, o programa DESENHA ficaria:

```
TO DESENHA :L
  SE :L > 100 [PARA]
  QUADRADO :L
  DESENHA :L + 10
FIM
```

Fica como sugestão fazer um programa que controla o número de quadrados.

9 Gravação de Programas

Para gravar os programas, é necessário dar nome à página do trabalho, utilizando o comando:

- NP “Meuprograma

Para gravar o arquivo **Meuprograma** em disco, basta teclar <Esc>. A gravação é total, isto é, a página gráfica e a de programação são gravadas.

10 Impressão

Existem os seguintes comandos para impressão:

- IMPRIMETEXTO, para imprimir o texto...
- IMPRIMETEXTO80, para imprimir texto em 80 colunas
- IMPRIMEECRAN, para imprimir texto e gráficos

Os comandos UMALINHA e DUASLINHAS são usados para imprimir textos com um e dois espaços, respectivamente, entre as linhas.

11 Exemplos de Programas em LOGO

Estes programas¹ foram implementados por três alunos da 6ª série do primeiro grau, em um trabalho que realizei, durante um ano (1992), sobre como ensinar conceitos de uma linguagem de programação de uma maneira simples.

1. Moinho — Este exemplo desenha um moinho.
Para executar digite "moinho"

```
to qua :11
  repete 4 [ pf :11 vd 90 ]
fim
```

```
to ret :11 :12
  repete 2 [ pf :12 vd 90 pf :11 vd 90 ]
fim
```

```
to tri :n
  vd 90
  pf :n
  ve 120
  pf :n
  ve 120
  pf :n
fim
```

```
to posicao1
  ve 90
  pf 15
  vd 90
fim
```

```
to posicao2
  pf 50
fim
```

```
to posicao3
  lc
  pt 40
  lb
fim
```

¹Os exemplos mostrados nesta seção foram implementados pelos alunos André, Lia e Sandra como parte do processo de aprendizagem do LOGO.

```
to posicao4
  ve 120
  pf 20
  lc
  vd 90
  pf 5
  lb
  ve 45
fim
```

```
to casa
  ret 10 20
  posicao1
  ret 40 50
  posicao2
  tri 40
fim
```

```
to ret2 :11 :12
  repete 2 [ pf :11 / 2 vd 90 pf :12 vd
    90 pf :11 / 2 ]
fim
```

```
to vela
  vd 90
  pf 10
  ve 90
  ret2 15 30
fim
```

```
to posicao5
  ve 90
  pf 10
fim
```

```
to velas :cont
  se :cont = 4 [ para ]
  vela
  posicao5
  velas :cont + 1
fim
```

```
to moinho
  le
  et
  posicao3
```

```

casa
posicao4
velas 0
fim

```

2. Ursinho — Este exemplo mostra um bichinho come-come emitindo som. Para rodá-lo digite "BELEZINHA".

```

to ursinho :n
  vd 90
  la
  repete :n 4 [som 349 1 fixaforma 7 pf
    3 espera 4 som 880 1 fixaforma 8 pf 3 espera 4 ]
  lb
fim

```

```

to rua :11
  et
  ve 90
  pf 320
  ve 90
  lc
  pf :11
  vd 90
  lb
  pf 320
  lc
  vd 90
  pf :11 / 2
  vd 90
  repete 63 [ of 2 1c of 3 1b ]
  mt
fim

```

```

to belezinha
  rua 30
  ve 90
  ursinho 212
fim

```

3. Circo — Este exemplo apresenta o desenho de um circo e mostra uma roda gigante usando recursos de animação. Para executar digite "TUDO".

```

to circ :11
  repete 360 [ pf :11 ve 1 ]
fim

```

```

to posicao1
  ta [ 0 1 2 3 ]
  lc
  fixaforma 5
  fixapos [0 0 ]
  ta 0 repete 90 [ pf 0.7 ve 1 ]
  ta 1 repete 180 [ pf 0.7 ve 1 ]
  ta 2 repete 270 [ pf 0.7 ve 1 ]
  ta [ 0 1 2 3 ] mt

```

fim

```

to roda
  circ 0.7
  paus
  posicao1
  repete 360 [ pf 0.7 ve 1 ]

```

fim

```

to paus
  ve 90
  pf 40
  vd 90
  repete 12 [ pf 40 pt 40 ve 30 ]

```

fim

```

to tri :11
  vd 90
  repete 3 [ ve 120 pf :11 ]

```

fim

```

to posicao2

```

```

  lc
  pt 50
  ve 90
  pf 12.5
  vd 90
  lb
  tri 55
  lc
  ec
  lb

```

fim

```

to parque
  posicao2

```

```

    roda
fim

to umquarto
    repete 90 [ pf 0.7 ve 1 ]
fim

to umquartot
    repete 90 [ pt 0.7 ve 1 ]
fim

to pintri :11 :n
    sesenao :11 >0
    [trie :11 pt :n ve 90 pintri :11 - 1 :n] [ 1c 1b ]
fim

to trie :11
    vd 90
    repete 3 [ ve 120 pf :11 ]
fim

to lona
    vd 90
    umquarto
    umquartot
    ve 90
    pf 60
    vd 90
    lc
    pf 80
    lb
    vd 90
    pf 60
fim

to bandeirinha
    repete 4 [trie 10 pt ve 90 pintri 10 0.5 ve 90 pf 5 vd 90 ]
fim

to circo
    lc
    pf 10
    lb
    lona
    vd 180
    bandeirinha

```

```
posicao3
red 30 25
fim
```

```
to posicao3
  lc
  pf 60
  vd 90
  pf 55
  vd 90
  lb
fim
```

```
to red :n1 :n2
  pf :n1
  vd 90
  pf :n2
  vd 90
  pf :n1
fim
```

```
to tudo
  chao
  lc
  vd 90
  pf 30
  ve 90
  lb
  circo
  lc
  ec
  lb
  parque
fim
```

```
to montanha :n
  repete 180 [ pf :n ve 1 ]
fim
```

```
to chao
  lc
  pt 50
  vd 90
  lb
  pf 320
  lc
```

```
ec
fim
```

```
to a :a
  sesenao :a > 0 [ es "a ] [ es "b ]
fim
```

4. **Cidade** — Este exemplo usa recursos de animação para desenhar as ruas de uma cidade. Para executar digite "PLANO2".

```
to estrada
  vd 90
  pf 320
  ve 90
  lc
  pf 30
  vd 90
  lb
  pf 320
  lc
  vd 90
  pf 15
  vd 90
  repete 20 [ pf 10 lc pf 10 lb ]
  vd 90
  lc
  pf 7.5
  ve 90
fim
```

```
to carros
  ta 0 fixaforma 28 lc ec pt 60 vd 90
  ta 1 fixaforma 27 lc pf 59 vd 90
  ta 2 fixaforma 1 lc pf 73 ve 90
  ta 3 fixaforma 3 lc pt 50 ve 90
  ta [ 0 1 2 3 ] mt
  lc
  repete 50 [ta 1 pf 5 espera 1 ta 0 pf
  10 espera 3 ta 2 pf 15 espera 3 som 40
  2 ta 3 anda 1 ]
fim
```

```
to calçada
  vd 90
  pf 320
  ve 60
```

repete 32 [pf 10 vd 120 pf 10 ve 120]
lc
ec
pf 50
vd 180
lb
fim

to plano1
calcada
calcada
lc
pf 10
lb
estrada
lc
fim

TO CASA :V
PF :V
VD 60
PF :V / 2
VD 60
PF :V /2
VD 60
PF :V
VE 90
PF 5
VE 90
FIM

TO CIDADE
LC
PF 55
LB
CASA 15
CASA 10
CASA 18
CASA 13
CASA 16
CASA 20
CASA 8
CASA 11
CASA 15
CASA 21
CASA 9

CASA 15
CASA 10
CASA 18
CASA 8
CASA 13
CASA 15
CASA 20
CASA 6
LC
EC
LB
FIM

TO PLANO2
et
CIDADE
ferro
lc
ec
lb
PLANO1
CARROS
FIM

to anda :p
repete :p [fixaforma 2 pf 3 fixaforma 3
espera 2 pf 3]
fim

to tri :v
ve 90
vd 60
pf :v
ve 120
pf :v
ve 120
pf :v
fim

to ret :11 :12
pf :11
vd 90
pf :12
vd 90
pf :11
vd 90

pf :12
fim

to ferro

lc

ec

pt 60

ve 90

pf 60

lb

ret 40 20

vd 90

pf 13.3

ret 5 10

vd 90

pf 26.6

vd 90

pf 20

ret 15 15

vd 90

pf 15

vd 90

pf 15

ve 90

tri 15

lc

vd 90

pf 40

vd 90

lb

pf 320

lc

ve 90

pf 10

lb

vd 90

pf 320

vd 90

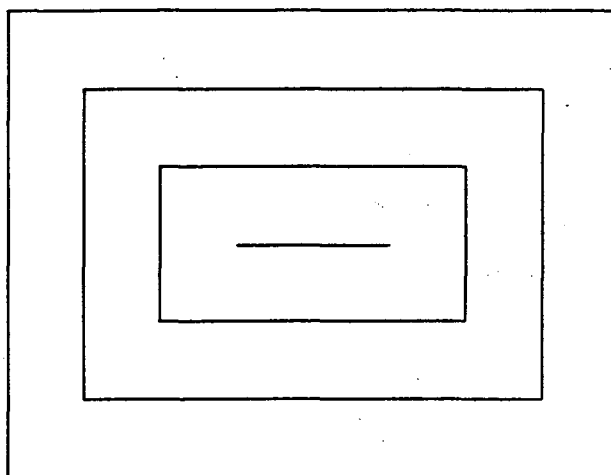
repete 35 [pf 10 vd 90 pf 5 vd 90 pf 10 ve 90 pf 5 ve 90]

fim

12 Exercícios

O objetivo destes exercícios é fazer uma revisão geral. Cabe, portanto, a cada aprendiz deixar a sua imaginação livre e tentar ver o que ele consegue criar com os comandos vistos em cada seção.

1. O que é LOGO?
2. O que são comandos básicos (primitivos)? Dê exemplos (pelo menos 10).
3. Qual é a forma (sintaxe) que é utilizada para escrever um programa em LOGO? O que é um programa? Construa um programa que faça o desenho de uma circunferência.
4. Observe o desenho:



- a) Como você faria um programa para fazer este desenho?
 - b) É possível fazer este programa usando outras primitivas, ou seja, comandos diferentes dos que foram usados na letra a)? Quais?
5. Qual o procedimento usado para desenhar uma nova forma de “tartaruga”?
 6. Suponha que você tem tartarugas no formato de dois homens, como você pode fazer um programa, alternando as formas, para fazer um homem caminhando?
 7. É mais interessante caminhar ouvindo música. Como você pode fazer isso no seu programa?
 8. Qual é a sintaxe dos comandos **repete** e **se** ? Qual a diferença entre eles, ou seja, onde você usa **repete** e onde usa **se**?

13 Conclusões

Como comentado anteriormente, esta Nota Didática tem como objetivo familiarizar o leitor com as noções básicas da linguagem de programação LOGO.

A publicação de programas LOGO que implementam problemas envolvendo recursão intenciona contribuir para um melhor entendimento desta técnica, que pode ser facilmente implementada nesta linguagem, usando recursos de visualização através da tartaruga.

Agradecimentos:

Gostaria de agradecer à Profa. Renata Pontim de Mattos e a Ana Cláudia Noivo Arantes pela leitura, correção e comentários sobre o conteúdo desta Nota.

Referências

[LOGOWRITER] Manuais do LOGOWRITER.

[Papert 80] Papert, S. *LOGO: Computadores e Educação* Editora Brasiliense, 1980.