

Softwares Livres Educativos



XLogo
Versão 0.9.92
Linguagem de Programação

Softwares Livres Educação



XLogo

Versão 0.9.92

Linguagem de Programação



SECRETARIA DE ESTADO DA EDUCAÇÃO
SUPERINTENDÊNCIA DA EDUCAÇÃO
DIRETORIA DE TECNOLOGIA EDUCACIONAL
COORDENAÇÃO DE MULTIMEIOS

XLogo
Versão 0.9.92
Linguagem de Programação

CURITIBA
SEED - PR
2010

É permitida a reprodução total ou parcial desta obra, desde que seja citada a fonte. Disponível também em: <<http://www.diaadia.pr.gov.br/multimeios>>

Coordenação de Multimeios

Organização

Arnoldo Brasílio Filho

Coordenação de Mídia Impressa e Web

Revisão ortográfica

Aquias da Silva Valasco

Bárbara Reis Chaves Alvim

Orly Marion Webber Milani

Tatiane Valéria Rogério de Carvalho

Coordenação de Multimeios

Produção

Eziquiel Menta

Ricardo Mendonça Petracca

Capa

Jocelin José Vianna da Silva

Rafael Cadilhe David

Projeto Gráfico

Juliana Gomes de Souza Dias

Diagramação

Taísa Delazeri Burtet

CATALOGAÇÃO NA FONTE - CEDITEC-SEED-PR

Paraná. Secretaria de Estado da Educação. Superintendência da Educação. Diretoria de Tecnologias Educacionais. P111

Xlogo : versão 0.9.92 ; linguagem de programação / Secretaria de Estado da Educação. Superintendência da Educação. Diretoria de Tecnologias Educacionais. – Curitiba : SEED – Pr., 2010. – 60 p – ISBN978-85-8015-028-5

1. Xlogo. 2. Linguagem de programação . 3. Educação-Paraná. I. Brasílio Filho, Arnoldo, org. II. Título.

CDD600

CDU 6+37(816.2)

SECRETARIA DE ESTADO DA EDUCAÇÃO

Superintendência da Educação

Diretoria de Tecnologia Educacional

Coordenação de Multimeios

Rua Salvador Ferrante, 1.651 – Boqueirão

CEP 81670-390 – Curitiba – Paraná

www.diaadia.pr.gov.br/multimeios

**IMPRESSO NO BRASIL
DISTRIBUIÇÃO GRATUITA**

APRESENTAÇÃO

A Diretoria de Tecnologia Educacional (Ditec), da Secretaria de Estado da Educação do Paraná, viabiliza ações que possibilitam não apenas o acesso operacional aos equipamentos existentes na escola, mas também a pesquisa, a produção e a veiculação de conteúdos educacionais de forma compatível com os avanços tecnológicos.

Nossa proposta consiste na integração e articulação das mídias com o mundo moderno por meio de ações desenvolvidas pela TV Paulo Freire, pelo Multimeios e pelo Portal Dia-a-dia Educação. Além do acesso à tecnologia, entendemos que é necessária a orientação para o seu uso. Essa é a tarefa da equipe dos assessores da Coordenação Regional de Tecnologia na Educação (CRTE) presente nos Núcleos Regionais de Educação e que atende aos professores das escolas da Rede Pública.

Para complementar a ação de produção e disponibilização de materiais didáticos de apoio ao uso de tecnologia, o Multimeios elaborou tutoriais de alguns dos *softwares* educacionais instalados no Paraná Digital para serem disponibilizados aos usuários dos laboratórios. Entre eles encontra-se o tutorial do XLogo, um *software* que, para alunos e professores, é um meio acessível e poderoso de criar simulações, apresentações, gráficos, textos, jogos e animações.

Acima de tudo, um recurso flexível que permite a construção de programas, o teste de hipóteses, a manipulação de variáveis e a reflexão sobre os próprios processos de aprendizagem, pois estimula o desenvolvimento de estratégias de raciocínio e a conscientização do próprio processo de aprendizagem.

O conteúdo deste tutorial foi autorizado por Alexandre R. Soares, responsável pela versão em português do Manual do usuário XLogo, disponível na página oficial do *software*, para uso da Secretaria de Estado da Educação do Paraná e adequações necessárias à realidade da Rede Pública Estadual de Ensino.

Elisabete dos Santos
Diretora de Tecnologia Educacional

Eziquiel Menta
Coordenador de Multimeios

SUMÁRIO

1 INTRODUÇÃO	8
2 INTERFACE	9
2.1 JANELA PRINCIPAL	9
2.2 EDITOR DE PROCEDIMENTOS	9
3 MENUS	11
3.1 MENU ARQUIVO.....	11
3.2 MENU EDIÇÃO.....	12
3.3 MENU FERRAMENTAS	12
3.4 MENU AJUDA	13
4 CONVENÇÕES ADOTADAS	13
4.1 COMANDOS E SUA INTERPRETAÇÃO	13
4.2 PROCEDIMENTOS	14
4.3 O CARACTERE ESPECIAL “\”	15
4.4 CAIXA ALTA	16
4.5 OPERADORES E SINTAXE	16
4.6 USO DE CORES.....	17
5 PRIMITIVAS	18
5.1 MOVER A TAT E CONFIGURAÇÕES.....	18
5.2 TAT NO ESPAÇO	23
5.2.1 A técnica da perspectiva	23
5.2.2 Primitivas para 2D e 3D.....	25
5.2.3 Primitivas exclusivas para o modo 3D	26
5.2.4 Visor 3D.....	27
5.3 ESCREVER.....	28
5.4 ARITMÉTICA E OPERAÇÕES LÓGICAS	29
5.5 OPERAÇÕES COM LISTAS	31
5.6 BOOLEANOS	32
5.7 TESTAR UMA EXPRESSÃO	33
5.8 LIDAR COM PROCEDIMENTOS E VARIÁVEIS.....	33
5.9 RASTREAR	34
5.10 ACESSAR ARQUIVOS	35
5.11 FUNÇÕES PINTE E PINTEZONA.....	38

6 RECEBER EM ENTRADA DE USUÁRIO.....	39
6.1 INTERAGIR PELO TECLADO.....	39
6.2 INTERAGIR COM O MOUSE.....	40
6.3 INTERAGIR POR IG (INTERFACE GRÁFICA)	43
7 TEMPO E DATA.....	46
8 EXEMPLOS DE PROGRAMAS	47
8.1 DESENHAR CASAS	47
8.2 DESENHAR UM RETÂNGULO PREENCHIDO	47
8.3 FATORIAL	48
8.4 FLOCO DE NEVE	48
8.5 USO DE LISTAS	49
8.6 UMA LINDA ROSÁCEA	49
9 INSTALAR, DESINSTALAR E ATUALIZAR.....	50
10 LISTA DE PRIMITIVAS	51
11 TRIGONOMETRIA.....	59
REFERÊNCIAS.....	60

1 INTRODUÇÃO

A linguagem Logo é largamente utilizada no ensino de Geometria, mesmo nas séries fundamentais, servindo como ferramenta no aprendizado de fundamentos de programação, construção e visualização de formas geométricas. Com essa linguagem, é possível desenvolver um ambiente de estudo de Geometria, utilizando comandos simples e intuitivos, como “PD = Para direita”.

O *software* XLogo utiliza a linguagem Logo para desenvolvimento de estruturas geométricas, é *software* livre e multiplataforma, ou seja, está disponível gratuitamente para todos os sistemas operacionais. Conceitos trabalhados para o professor, como perspectivas e 3D, são trabalhados no XLogo utilizando comandos para “desenhar os objetos”, tornando o estudo da Geometria interessante e proporcionando uma aprendizagem mais significativa dos conceitos.

A proposta deste tutorial¹ é apresentar o *software* XLogo por meio de exercícios e exemplos que podem ser utilizados no laboratório de informática com os alunos, trabalhando os principais conceitos geométricos por meio de exemplos ilustrativos, com o passo a passo e comentários sobre as linhas de programação.

1 XLogo: Manual do usuário. A versão original deste tutorial foi gerada utilizando LaTeX2HTML translator Versão 2002-2-1 (1.70). Copyright © 1993, 1994, 1995, 1996, Nikos Drakos, Computer Based Learning Unit, University of Leeds. Copyright © 1997, 1998, 1999, Ross Moore, Mathematics Department, Macquarie University, Sydney. Argumentos de linhas de comando usados: latex2html -local_icons manuel-xlogo-en.tex

Tradução para o inglês iniciada por Loïc em 15 fev. 2005. Tradução para o português do Brasil, iniciada por Alexandre R. Soares em 04 abr. 2005.

Visite as páginas do XLogo: <<http://xlogo.tuxfamily.org>>.

2 INTERFACE

2.1 JANELA PRINCIPAL

No topo da Janela Principal, encontram-se os menus de costume: **Arquivo**, **Edição**, **Ferramentas** e **Ajuda**. Logo abaixo aparece a linha de comando onde são inseridos os comandos (instruções). Para a instrução ser executada, basta dar *Enter*. No meio, temos a área de desenho e na parte de baixo, encontra-se o histórico de comandos. Ele mostra as instruções já executadas, bem como as respectivas respostas. Eventuais mensagens de erro aparecem em vermelho. Para chamar um comando já executado, basta clicar sobre ele no histórico de comandos ou utilizar as setas de teclado (para baixo e para cima) na linha de comando. No canto inferior direito há dois botões: **Editor** e **Pare**. O primeiro interrompe a execução do programa e o segundo abre a janela do **Editor de procedimentos**.



Figura 1 – Janela principal do XLogo

2.2 EDITOR DE PROCEDIMENTOS

Há três modos de abrir o editor:

- escrevendo “ed” na linha de comandos (na parte superior da tela). O editor abre para exibir todos os procedimentos já definidos. Se desejar editar apenas alguns dos procedimentos, escreva: ed [procedimento_1 procedimento_2 ...];

- pressionando o botão **Editor** no canto inferior direito da janela do XLogo;
- utilizando as teclas de atalho *Alt+E*.

Estes são os botões que você encontrará no **Editor**:

 ou 	Fecha o editor e guarda (salva) o que foi feito. Este é o botão que deve ser pressionado sempre que escrever novos comandos ou fizer alterações nos procedimentos. Se preferir, utilize as teclas de atalho ALT+Q .		Este botão aparece na base do editor com uma caixa de texto para definir um comando principal, ou seja, o comando que será executado ao clicarmos no botão correspondente da janela principal. Este comando principal será salvo junto com o arquivo .lgo .
	Sai do editor sem salvar nenhuma das mudanças feitas. Você também pode utilizar as teclas de atalho ALT+C .		Recorta o texto selecionado para o clipboard.
	Imprime o conteúdo do editor.		Cola o texto armazenado no clipboard.
	Copia o texto selecionado para o clipboard.		Localiza e substitui termos no editor.

Importante: note que clicar no botão de fechar (x), no canto superior direito da janela de título, não produz efeito algum. Somente os dois primeiros botões acima citados permitem sair do editor. Para apagar (eliminar) procedimentos, utilize as primitivas **elimine** e **eliminetudo** (et).

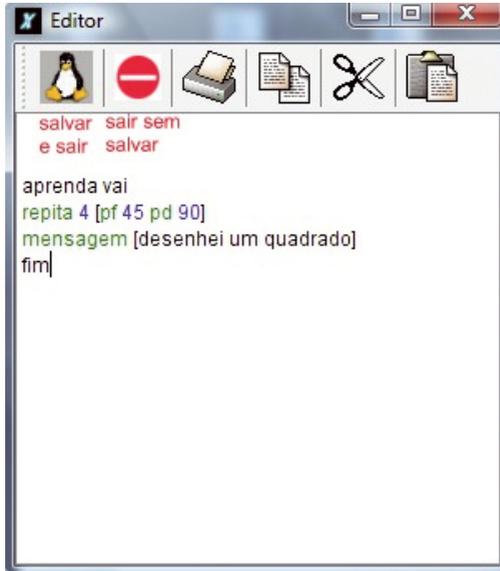


Figura 2 – Janela do Editor de Procedimentos

Desde a versão 0.9.93, o ícone do pinguim foi substituído por um triângulo (o mesmo utilizado para representar a Tat em diversos programas de linguagem Logo).

3 MENUS

3.1 MENU ARQUIVO

Arquivo / Abrir: abre (carrega) algum arquivo Logo já existente;

Arquivo / Guardar: salva os procedimentos do arquivo em uso;

Arquivo / Guardar como: salva os procedimentos atuais em um arquivo com outro nome;

Arquivo / Sair: fecha o XLogo;

Arquivo / Capturar imagem / Guardar imagem como: salva o desenho exibido na tela do XLogo como uma imagem nos formatos JPG ou PNG. Se desejar gravar somente uma parte do desenho, utilize o *mouse* conforme as instruções que aparecerão numa caixa de diálogo (um clique no canto superior esquerdo, outro no canto inferior direito);

Arquivo / Capturar imagem / Imprimir Imagem: imprime o desenho exibido na tela do XLogo. Também é possível imprimir somente parte do desenho da mesma forma descrita no item anterior.

3.2 MENU EDIÇÃO

Edição / Copiar: copia o texto selecionado para o clipboard;

Edição / Recortar: recorta o texto selecionado para o clipboard;

Edição / Colar: cola o texto armazenado no clipboard na linha de comando.

3.3 MENU FERRAMENTAS

Ferramentas / Mudar cor do lápis: seleciona uma cor para o lápis da tartaruga. Também é possível fazê-lo com o comando **mudecl**;

Ferramentas / Mudar cor do fundo: seleciona uma cor para o fundo da tela. Também é possível fazê-lo com o comando **mudecf**;

Ferramentas / Arquivo inicial: informa o caminho para os arquivos de inicialização. Qualquer procedimento contido em arquivos no formato ***.lgo** se tornarão “pseudoprimitivas” para o XLogo (algo como os procedimentos de biblioteca do SuperLogo ou MSWLogo). Elas não poderão ser editadas ou alteradas pelo usuário. Desse modo, você pode criar primitivas personalizadas;

Ferramentas / Traduzir procedimentos: traduz automaticamente qualquer procedimento de um idioma para outro. Pode ser útil, por exemplo, ao copiar da Internet algum projeto em outro idioma;

Ferramentas / Apaga procedimentos: seleciona os procedimentos que deseja eliminar. Veja a primitiva **elimine**;

Ferramentas / Preferências: abre uma caixa de diálogo na qual você pode configurar:

- **idioma:** francês, espanhol, inglês ou português. Note que as primitivas são diferentes em cada idioma;

- **aparência:** seleciona uma aparência para o seu XLogo (Metal, Windows e Motif são as opções);
- **velocidade da tartaruga:** se desejar ver todos os movimentos da tat, você pode torná-la mais lenta utilizando a barra deslizante.

Na segunda aba, você pode escolher um “roupa para a tat” (uma figura para a tartaruga). Nas abas seguintes, você pode controlar número de tartarugas, estilo de lápis, instrumentos de som (mudeinstrumento número) e fonte (tipo e tamanho).

3.4 MENU AJUDA

Menu / Licença: exibe a licença pública geral (GPL) sob a qual este programa é distribuído;

Menu / Tradução da licença: exibe uma tradução da licença acima. Essa tradução não é oficial (apenas para a versão em inglês) e está aqui apenas para ajudar no entendimento;

Menu / Sobre: o “de sempre” e mais o endereço <http://xlogo.tuxfamily.org> para o marcador do seu navegador (bookmarks).

4 CONVENÇÕES ADOTADAS

Esta seção aborda alguns pontos essenciais da linguagem Logo e também do XLogo.

4.1 COMANDOS E SUA INTERPRETAÇÃO

A linguagem Logo permite que certos eventos sejam disparados por comandos internos, chamados de primitivas. Cada primitiva pode ter um certo número de parâmetros, os quais são chamados argumentos. Por exemplo, a primitiva **ld**, que limpa a tela de desenho, não exige nenhum argumento, enquanto que a primitiva **soma** exige dois argumentos.

mostre soma 2 3 retornará 5.

Há três tipos de argumentos no Logo:

- **números:** algumas primitivas exigem números como argumento; **pf 100** é um exemplo;
- **palavras:** são marcadas por aspas duplas (“”). Um exemplo de primitiva com argumento é **mostre**;

mostre “Olá devolve Olá.

Note que se você esquecer o (“), o interpretador retornará uma mensagem de erro. De fato, **mostre** espera um argumento, ou, para o interpretador, “Olá” não significará nada, uma vez que não é um número, uma palavra, uma lista ou qualquer outra coisa definida em um procedimento.

- **listas:** são definidas entre colchetes.
- **números:** são tratados em algumas instâncias como um valor numérico (por exemplo, **pf 100**) e, em outras, como uma palavra (por exemplo: **mostre évazio? 12** escreverá falso).

4.2 PROCEDIMENTOS

Além das primitivas já existentes, você pode criar seus próprios comandos. Eles são chamados procedimentos.

Procedimentos começam pela palavra “aprenda” e terminam com a palavra “fim”. Eles podem ser criados utilizando o editor interno do Xlogo (clique no botão **Editor**, no canto direito inferior do programa).

Eis um exemplo simples:

```
aprenda quadrado  
repita 4[parafrente 100 paradireita 90]  
fim
```

Agora escreva a palavra “quadrado” na caixa de comandos e dê **Enter**. A tat desenhará um quadrado. Os procedimentos também podem trabalhar com argumentos. Para isso, utilizamos variáveis. Uma variável é uma palavra que pode assumir um valor. Eis um exemplo simples:

```

aprenda dobro: palavra
mostre: palavra
mostre: palavra
fim
dobro [1 2 3] -----> 1 2 3
1 2 3
dobro "viva -----> viva
viva
dobro [viva a vida] -----> viva a vida
viva a vida

```

Veja outros exemplos de procedimentos neste manual.

4.3 O CARACTERE ESPECIAL “\”

O caractere “\” (barra invertida, ou backlash) permite, em particular, colocar espaço entre palavras, para uma nova linha, “\n” faz a quebra de linha enquanto “\”, seguido de um espaço, indica espaço numa palavra.

Exemplo:

```

mo "xlogo\ xlogo
xlogo xlogo
mo "xlogo\nxlogo
xlogo
xlogo

```

Se desejar escrever o caractere “\”, será necessário escrevê-lo duas vezes (“\\”).

```

mo "\\xlogo
\xlogo
mo "\\

```

Da mesma forma os caracteres “() [] #” são reservados para a linguagem Logo. Sendo assim, se necessitar representá-los, bastará um caractere “\” antes.

```

mo "(xlogo)
(xlogo)

```

Todos os caracteres “\” são ignorados. Esse aspecto é especialmente importante para o gerenciamento de arquivos.

No Windows, por exemplo, para definir (mudar) o diretório atual para X:\Meus Documentos, escreva:

```
mudediretório "c:\\Meus\ Documentos
```

Já no Linux, para definir (mudar) o diretório atual para /local/aluno/Meus Documentos, escreva:

```
mudediretório "/local/aluno/Meus\ Documentos
```

Note o uso do “\” para indicar o espaço entre “Meus” e “Documentos”. No exemplo do Windows, se você omitisse a dupla barra invertida, o caminho seria definido como C:Meus Documentos e o XLogo devolveria uma mensagem de erro (diretório inválido).

4.4 CAIXA ALTA

O XLogo não faz distinção entre maiúsculas e minúsculas para procedimentos e primitivas. Assim, no procedimento quadrado, definido anteriormente, tanto faz escrever QUADRADO ou qUAdRAdO; o interpretador de comandos o “traduzirá” e executará corretamente. Por outro lado, XLogo faz distinção para listas e palavras:

```
mostre "BelEzA ----> "BelEzA (as letras maiúsculas e minúsculas são mantidas como escritas)
```

4.5 OPERADORES E SINTAXE

Há dois modos de escrever certos comandos. Por exemplo, para somar 4 e 7, há duas possibilidades: você tanto pode utilizar a primitiva **soma**, que exige dois argumentos **soma 4 7**, ou você pode utilizar o operador +: 4+7. Ambos têm o mesmo efeito. A tabela a seguir mostra a equivalência entre operadores e primitivas:

Soma +	Diferença -	Produto *	Quociente /	Ou 	E &	São iguais? =
soma 4 7	diferença 4 7	produto 4 7	quociente 4 7	3 = 4 ou (7 <= 49/7)	3 = 4 ou (7 = 49/7)	3 + 4 = 7 - 1
4 + 7	4 + 7	4 + 7	4 / 7	3=4 (7 <= 49/7)	3=4 & (7 = 49/7)	

Os operadores “|” e “&” são específicos do XLogo. Eles não existem nas versões tradicionais de Logo. Desde a versão 0.9.92, temos dois novos operadores (sem primitivas associadas):

<= (“menor que”)

>= (“maior que”)

Eis alguns exemplos de seus usos:

mo 3+4=7-1 ----> verd

mo 3=4 | (7 <= 49/7) ----> verd

mo 3=4 & 7=49/7 ----> falso

4.6 USO DE CORES

As cores são definidas no XLogo com a ajuda de três números entre 0 e 255. Este é o sistema de código RGB (red, green, blue). Cada número corresponde respectivamente a uma intensidade de vermelho (red), de verde (green) e de azul (blue) para a cor considerada. Uma vez que este sistema não é intuitivo, o XLogo proporciona 16 cores predefinidas acessíveis por um número ou um nome.

Número	Primitiva	[R G B]	Cor	Número	Primitiva	[R G B]	Cor
0	preto	[0 0 0]		9	cinzaclaro	[192 192 192]	
1	vermelho	[255 0 0]		10	vermelhoescuro	[128 0 0]	
2	verde	[0 255 0]		11	verdeescuro	[0 128 0]	
3	amarelo	[255 255 0]		12	azulescuro	[0 0 128]	
4	azul	[0 0 255]		13	laranja	[255 200 0]	
5	magenta	[255 0 255]		14	rosa	[255 175 175]	
6	ciano	[0 255 255]		15	violeta	[128 0 255]	
7	branco	[255 255 255]		16	marrom	[153 102 0]	
8	cinza	[128 128 128]					

Observe os três comandos que darão o mesmo efeito:

mudcf laranja

mudcf 13

mudcf [255 200 0]

5 PRIMITIVAS

A tartaruga (“tat”) é controlada por comandos chamados primitivas. Elas são apresentadas nas seções a seguir.

5.1 MOVER A TAT E CONFIGURAÇÕES

Nesta primeira tabela, você encontra as primitivas que coordenam os movimentos da tartaruga.

Primitivas	Argumento	Uso
paradireita, pd	n: ângulo	Gira a tat n graus para a direita em relação à direção que ela está apontando.
paraesquerda, pe	n: ângulo	Gira a tat n graus para a esquerda em relação à direção que ela está apontando.
parafrente, pf	n: número de passos	Move a tat para frente n passos na direção que ela está apontando.
paratrás, pt	n: número de passos	Move a tat para trás n passos na direção que ela está apontando.

arco	a b c	arco a b c desenha um arco de círculo de a passos ao redor da tat entre os ângulos b e c dela (a tat é o centro do círculo). Exemplo: arc 200 20 80.
carregueimagem, carimg	a: lista	Carrega uma imagem na tela do XLogo. A posição da tat será o canto superior esquerdo da imagem. São aceitas apenas imagens .png e .jpg . O caminho não pode conter espaços e ser absoluto (não relativo), ou seja, tem que ser completo desde o topo da árvore. Por exemplo: carimg [C:\\diretório_das_minhas_imagens\turtle.jpg].
centro	nenhum	Coloca a tat na posição inicial, isto é, na origem (coordenadas [0 0]) e com direção 0 grau (aponta para cima na tela).
círculo, circ	R: raio	Desenha um círculo de raio R (a tat é o centro do círculo).
mudedireção, mudedç	n: orientação	Orienta a tartaruga para a direção especificada. 0 (zero) corresponde à tat apontada para cima na tela. A orientação da tartaruga corresponde aos valores lidos em um transferidor.
mudepos	[x y]: lista com 2 números	Move a tat para as coordenadas especificadas pelos dois números na lista (x especifica o eixo x , e y o eixo y).
mudex	x: eixo x	Move a tat horizontalmente para o ponto x no eixo x .
mudexy	x y: abscissa x seguida pela ordenada y	Idêntica à mudepos [x y] (a diferença é que os valores não estão em uma lista, ou seja, não estão entre colchetes).
mudey	y: eixo y	Move a tat verticalmente para o ponto y no eixo y .
ponto	a: lista	Coloca um ponto nas coordenadas indicadas.
rotule	a: palavra ou lista	Desenha uma palavra ou lista a partir da posição da tat na mesma orientação (mudedireção). Por exemplo: rotule [Que beleza!] escreverá a sentença "Que beleza!" onde quer que a tat esteja.
tamanhorotule, tr	a: palavra ou lista	Retorna o comprimento (em passos de tartaruga) necessário para escrever a palavra ou a lista desejada na área de desenho utilizando a fonte selecionada.

Esta segunda tabela mostra as primitivas que ajustam as propriedades da tat. Por exemplo, a tartaruga está visível na tela? Que cor está em uso pelo lápis?

Primitivas	Argumentos	Uso
animado	verd ou falso	Passa ao modo animado. A tat não desenha mais na área gráfica, apenas na memória. Para transferir o desenho para a tela, utilize a primitiva veranimado. Muito útil para criar animações ou efetuar um desenho mais rapidamente.
cercar	nenhum	A tat é confinada à área de desenho. Aparecerá uma mensagem de erro se a tat for movida para além do campo de desenho e informará o número máximo de passos que a tat ainda pode dar até o limite de sua área.
comlimite	nenhum	A tat não pode ultrapassar o campo de visão na tela; ela reaparece no lado oposto.
cordofundo, cf	a: lista	Informa a cor de fundo (da tela). A cor é indicada por uma lista [R G B] em que R é "vermelho", G é "verde" e B é "azul".
cordolápis, cl	a: lista	Informa a cor do lápis em uso. A cor é indicada por uma lista [R G B] em que R é "vermelho", G é "verde" e B é "azul".
cordoponto, cdp	a: lista	Informa a cor do pixel no ponto a . A cor é indicada por uma lista [R G B] em que R é "vermelho", G é "verde" e B é "azul".
direção, dç	nenhum	Informa a direção da tat (confira mudedireção).
direçãopara, dçpara	a: lista	A lista deve conter dois números (coordenadas). Informa o valor a ser utilizado por mudedireção para que a tat aponte para as coordenadas indicadas nessa lista.
distância, dist	a: lista	A lista deve conter dois números representando as coordenadas. Informa o número de passos entre a posição atual e a indicada pelas coordenadas na lista.
eixo	n: inteiro	Traça dois eixos com espaçamento n (passos de tartaruga). Contrário de semeixo.
eiox	n: inteiro	Traça eixo horizontal com espaçamento n (passos de tartaruga). Exemplo: eiox 30.
eioy	n: inteiro	Traça eixo vertical com espaçamento n (passos de tartaruga).
eixo?	nenhum	
eiox?	nenhum	Informa verd (o eixo está em exibição) ou falso.
eioy?	nenhum	

escondetata, dt	nenhum	Torna a tática invisível na tela.
fonte	nenhum	Devolve o tamanho da fonte a usar pela primitiva rotule.
grade	a b	Exibe uma grade na área de desenho com quadrículas de largura a e altura b . Para removê-la, utilize semgrade. Exemplo: <code>grade 10 10</code> .
grade?	nenhum	Devolve <code>verd</code> (a grade está em exibição) ou <code>falso</code> (a grade não está em exibição).
invertelápis, il	nenhum	A tática utiliza lápis no modo inverso, isto é, riscar onde não tiver nada e apagar se já tiver.
lápis pinta, lp	nenhum	A tática utiliza lápis para riscar com sua cor clássica (preta).
limpedesenho, ld	nenhum	Limpa todos os desenhos na tela e restaura a tática (coloca-a no centro).
lt, limpetexto	nenhum	Limpa (apaga) tudo que estiver escrito na linha de comando e no histórico.
mensagem, msg	a: lista	Exibe uma caixa de diálogo com a mensagem escrita na lista. O programa é interrompido até que o usuário clique no botão OK .
mostretata, at	nenhum	Torna a tática visível na tela.
mudecordoeixo	a: número inteiro ou lista [verm verde azul]	Muda a cor do eixo segundo a convenção de cores abaixo (mudecordofundo): <code>mudecordoeixo [125 64 23] eixo 30</code> .
mudecordofundo, mudecf		0: preto; 1: vermelho; 2: verde; 3: amarelo; 4: azul; 5: magenta; 6: ciano; 7: branco. Para valores acima de 7, a escala de cores se repete.
mudecordograde, mudecdg		Muda a cor da grade (veja acima, em mudecordofundo, para a convenção de cores). Exemplo: <code>grade 30 30 mudecordograde [125 64 23]</code> .
mudecordolápis, mudecl		Muda a cor do lápis (mesma convenção de mudecordofundo).
mudeespesurado-lápis, mudeel	n: número	Define a espessura do lápis em <i>pixels</i> . O valor padrão é 1. A ponta do lápis é um quadrado (pode ser alterado para circular no menu Ferramentas, Preferências).
mudefonte, mundef	n: número	Ao utilizar a primitiva rotule para escrever na área de desenho, é possível modificar o tamanho da fonte com mudefonte. O tamanho padrão da fonte é 12.

mudnomefonte, mudenf	n: número	Escolha um número de fonte a usar pela primitiva rotule. A equivalência entre número e tipo de fonte está no menu Ferramentas / Preferências / Guia Fonte .
mudepontadolápis, mudepl	0 ou 1	Altera a ponta do lápis. 0 (quadrada) e 1 (redonda).
mudequalidadedai- magem, mudefi	0 ou 1 ou 2	Altera a qualidade do desenho. 0 (normal), 1 (alta) e 2 (baixa).
muderoupa	n: número	Escolha sua tat preferida na segunda guia do menu Ferramentas / Preferências.... ou ainda, utilize a primitiva muderoupa com números entre 0 e 6 (0 é um triângulo).
mudeseparação	a: número	Determina a proporção entre a janela de desenho e a janela de histórico de comandos. O número a é um valor entre 0 e 1. Se valer 1, a janela de desenho ocupará toda a área. Se valer 0, a janela de histórico é que ocupará toda a área.
mudetamanhodajane- la, mudetamjan	lista	Altera as dimensões da área de desenho. Exemplo: mudetamanhodajanela [650 850].
nomefonte	nenhum	Devolve uma lista com dois elementos. O primeiro é o número que corresponde à fonte em uso; o último é uma lista contendo nome da fonte.
pareanimado	nenhum	Interrompe o modo animado (veja esta primitiva) e volta ao modo clássico. A tat volta a desenhar na área gráfica, apenas na memória. A animação também pode ser interrompida clicando na figura de câmera que aparece à esquerda da janela de comandos já escritos.
pontadolápis, pl	nenhum	Exibe a forma da ponta do lápis. 0 (quadrada) e 1 (redonda).
pos	nenhum	Informa a posição da tat. Por exemplo: pos devolve [10 -100].
qualidadedaima- gem, qi	nenhum	Devolve a qualidade do desenho. 0 (normal), 1 (alta) e 2 (baixa).
roupa	nenhum	Informa o número que representa a roupa da tat.
semeixo	nenhum	Elimina o(s) eixo(s) da área de desenho criado(s) por eixo , eiox ou eioxo . Exemplo: eixo 30.
semgrade	nenhum	Elimina a grade da área de desenho. Experimente grade 10 10.
sem limite	nenhum	A tat pode ultrapassar o campo de visão da tela.

separação	a: número	Devolve a proporção entre as janelas de desenho e de histórico de comandos.
tamhodajanela, tamjan	nenhum	Devolve uma lista informando as dimensões da área de desenho.
tamanhojanela, tj	nenhum	Devolve uma lista formada pelas coordenadas do canto superior esquerdo da área de desenho e do canto direito inferior.
useborracha, ub	nenhum	A tat apagará o que ela encontrar ao passar por cima.
uselápis, ul	nenhum	A tat riscará a tela ao se mover.
usenada, un	nenhum	A tat não riscará a tela ao se mover.
veranimado	nenhum	No modo animado, a imagem será atualizada na área gráfica.
zoom	a	Amplia (ou reduz) a área de desenho de acordo com o fator a escolhido. Exemplo: zoom 2.5

5.2 TAT NO ESPAÇO

Desde a versão 0.9.92, a tat pode escapar do plano para deslocar-se no espaço. Para isso, utilize a primitiva perspectiva (ou logo3d). Bem-vindo ao mundo da perspectiva 3D!

5.2.1 A técnica da perspectiva

Para representar o espaço em três dimensões quando temos apenas duas (um plano), fazemos uso da perspectiva de projeção. Uma câmera observa a cena 3D e a visão é projetada sobre um plano intermediário. Observe o esquema abaixo:

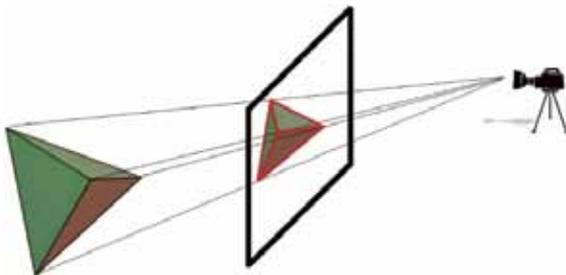


Figura 3 – Técnica da perspectiva

Certas primitivas nos permitem posicionar a câmera ao nosso modo, com a tela de projeção numa distância igual a metade da distância da câmera-origem do marcador.

No plano, a direção da tat está definida apenas para onde aponta seu “nariz” no espaço, a orientação da tat é dada por três valores de ângulos:

- **Rolagem (Roulis ou Roll):** giro longitudinal da tat (eixo $0y$);
- **Nariz (Tangage ou Pitch):** giro da tat sobre sua largura (eixo $0x$);
- **Direção Z (Cap pi Yaw):** giro da tat em torno do eixo $0z$.
- -0

Para tentar entender melhor, faremos uma comparação com os giros de manobras de um avião nas figuras abaixo (as denominações entre parênteses que aparecem nos tópicos acima são os nomes dessas manobras em francês e inglês, respectivamente):



Figura 4 –
Analogias
com avião

Pode parecer complexo num primeiro momento, mas você poderá notar que muitas coisas continuam as mesmas (ou parecidas) no mundo do plano 2D ou 3D. Eis aqui o que fazem as primitivas básicas para nos movermos no mundo 3D:

- **parafrente (pf) e paratrás (pt):** desloca a tat do mesmo jeito que no plano 2D;
- **paradireita (pd), paraesquerda (pe):** gira a tat como no plano 2D;
- **roleparadireita (rolepd):** a tat gira n graus para a direita em seu eixo longitudinal (Figura 4A)
- **roleparaesquerda (rolepe):** a tat gira n graus para a esquerda em seu eixo longitudinal (Figura 4A);
- **levantenariz:** a tat ergue seu nariz n graus, ou seja, em seu eixo transversal (Figura 4B);

- **baixenariz:** a tat gira seu nariz n graus para baixo, ou seja, em seu eixo transversal (Figura 4B).

No plano 2D, quando queremos desenhar um quadrado de 100 passos de lado, utilizamos:

```
repita 4 [fd 100 rt 90]
```

Essas instruções ainda são válidas no mundo 3D e o quadrado pode ser desenhado em perspectiva. Se a tat baixar o nariz 90 graus, poderemos desenhar outro quadrado:

```
ld
```

```
repita 4 [pf 100 pd 90]
```

```
baixenariz 90
```

```
repita 4 [pf 100 pd 90]
```

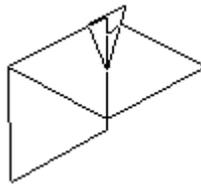


Figura 5 –
Quadrados em
perspectiva

À medida que você for experimentando alguns exemplos, irá entender mais e mais a orientação 3D e tornar-se um especialista!

Entenda que as três primitivas de rotação são interligadas. Tente o seguinte:

```
ld
```

```
roleparaesquerda 90 levantenariz 90 roleparadireita 90
```

5.2.2 Primitivas para 2D e 3D

As seguintes primitivas podem ser utilizadas tanto no plano quanto em 3D. A única diferença seriam os argumentos recebidos por essas primitivas. Por exemplo, a primitiva **mudepos** continua a esperar por uma lista de argumentos, mas desta vez precisaria de uma lista contendo três números (x ; y ; z), os quais representam os três pontos coordenados. Observe todas as primitivas:

Círculo	Arco	Centro	Direção para
distância	mudepos	mudex	mudey
mudedireção	rotule	tamanhorotule, tr	ponto
pos, posição	direção		

5.2.3 Primitivas exclusivas para o modo 3D

- **mudexyz:** desloca a tat até o ponto escolhido. Essa primitiva exige três argumentos que representam as coordenadas do ponto. É similar a **mudepos**, exceto por as coordenadas não serem fornecidas em uma lista. Exemplo: **mudexyz -100 200 50** moverá a tat para o ponto $x = -100$; $y = 200$; $z = 50$;
- **mudez:** desloca a tat pelo eixo z de modo equivalente a **mudex** e **mudey**;
- **mudeorientação:** muda a orientação da tat. Essa primitiva exige uma lista contendo três números (os ângulos de rolagem, levanta/baixa nariz e direção da cabeça no eixo z).
- **Exemplo:** com **mudeorientação [100 10 58]**, a tat rolará 100 graus, o nariz ficará 10 graus para cima (nariz empinado) e a cabeça apontará 58 graus para a direita.
- **orientação:** devolve a orientação da tat em uma lista do tipo (rolagem nariz cabeça).

Preste atenção nesta ordem de números: por exemplo, se a lista de valores for [100 20 90], significa que você deseja a mesma orientação a partir da posição original (após um limpdesenho), você deverá escrever a seguinte sequência: **roleparadireita 100 levantar nariz 20 paradireita 90**. Se você inverter a ordem desta instrução, não obterá a orientação desejada.

- **mudedireçãorolagem:** a tat gira em torno de seu eixo longitudinal até atingir o ângulo indicado;
- **direçãorolagem:** informa o valor em graus da rolagem atual;
- **mudedireçãonariz:** a tat gira em torno de seu eixo

transversal, apontando o nariz tantos graus para cima ou para baixo;

- **direçãonariz:** informa o valor em graus da direção atual do nariz (erguido ou baixado).

5.2.4 Visor 3D

O XLogo possui um visor 3D desde a versão 0.9.92, que permite visualizarmos desenhos 3D. Esse módulo utiliza bibliotecas Java3D, de modo que é necessário que se tenha o Java 3D completamente instalado.

Eis as regras a respeito de seu uso:

- Ao criarmos uma figura geométrica na área de desenho, temos que indicar ao visor 3D as formas que desejamos registrar para uma futura visualização.

É possível armazenar polígonos (superfícies), linhas, pontos ou textos. Para utilizar esse recurso, temos as seguintes primitivas:

- **iniciepolígono:** fará com que os movimentos seguintes da tat sejam utilizados para criar um polígono;
- **terminepolígono:** encerra o polígono iniciado por **iniciepolígono**. O polígono desenhado poderá ser exibido pelo visor 3D;
- **inicielinha:** fará com que os movimentos seguintes da tat sejam utilizados para criar uma linha representável no visor 3D;
- **terminelinha:** finaliza a linha iniciada por **inicielinha**. Essa linha é representável no visor 3D;
- **inicieponto:** fará com que os movimentos seguintes da tat sejam representados como um ponto no visor 3D;
- **termineponto:** finaliza o ponto iniciado por **inicieponto**;
- **inicietexto:** permite que seja representado no visor 3D qualquer texto escrito na área de desenho com a primitiva **rotule**;

- **terminetexto:** finaliza o texto iniciado por inicietexto para ser representado no visor 3D;

visor3d ou mostrepolígono: exibe o visor3d.

Por meio das primitivas apresentadas acima serão exibidos todos os objetos gravados na memória do XLogo.

Você poderá controlar o movimento da câmera com as seguintes opções:

- a cena será rotacionada ao se mover o *mouse* com o botão esquerdo pressionado;
- a cena será modificada ao se mover o *mouse* com o botão direito pressionado;
- o botão de rolagem do *mouse* será utilizado para controlar a ampliação da cena (*zoom*).

5.3 ESCREVER

Escreve na área de texto com as primitivas **mostre** ou **escreva**. Na tabela a seguir estão organizadas as primitivas que permitem ajustar as propriedades de texto. Tais primitivas controlam a cor e o tamanho na área de histórico e estão disponíveis somente para as primitivas **mostre** ou **escreva**.

Primitivas	Argumentos	Uso
cdt, cordotexto	nenhum	Devolve a cor da fonte no histórico de comandos.
escreva, esc	palavra, lista ou número	Idêntica à primitiva mostre, porém sem retorno de linha. Exemplo: esc "almo escreva "fada. Compare com mo "almo mo "fada.
estilo	nenhum	Devolve uma lista composta pelos diferentes estilos em uso pela primitiva mostre.
ftexto, fontetexto	nenhum	Devolve o tamanho da fonte.
lt, limpetexto	nenhum	Limpa o histórico de comandos.
mostre, mo	palavra, lista ou número	Idêntica à primitiva escrita, porém com retorno de linha. Exemplo: esc "almo escreva "fada. Compare com mo "almo mo "fada.
mudect, mudecordotexto	a: número ou lista	Define a cor da fonte no histórico de comandos.

mudeestilo	lista ou palavra	Muda o estilo em uso pela primitiva mostre. Os diferentes estilos possíveis são: nenhum, negrito, itálico, riscado, subscrito, sobrescrito ou sublinhado.
mudeft, mudefontetexto	a: número	Define o tamanho da fonte no histórico de comandos.
mudenft, mudenomefontetexto	n: número	Seleciona o número da fonte n ao escrever no histórico de comandos. A equivalência de número e fonte encontra-se no menu Ferramentas / Preferências / Guia Fonte .
nft, nomefontetexto	nenhum	Devolve uma lista com dois elementos. O primeiro é o número que corresponde à fonte utilizada no histórico de comandos. O último elemento é uma lista contendo o nome da fonte.

Alguns exemplos de formatação de texto com a primitiva **mostre**:

mudeestilo [negrito sublinhado] mostre "beleza

beleza

mudeestilo "riscado escreva [meu texto] mudestilo "itálico escreva "x

mudeestilo "sobrescrito escreva 2 meu texto x2.

5.4 ARITMÉTICA E OPERAÇÕES LÓGICAS

Eis uma lista de operadores lógicos:

Primitivas	Argumentos	Uso	Exemplos
ou	predicado1 predicado2	Retorna verd se uma das entradas for verdadeira. Caso contrário, retorna falso . Todas as entradas (predicados) precisam retornar verd ou falso .	mo ou 2 > 3 2 < 3
e	predicado1 predicado2	Retorna verd se os parâmetros de entrada forem verdadeiros. Caso contrário, retorna a palavra falso . Todas as entradas (predicados) precisam retornar verd ou falso.	mostre (e [a]=[a] [a]=[b])

não	predicado	Retorna verd se os parâmetros de entrada forem falsos. Caso contrário, retorna a palavra verd . A entrada (predicado) precisa retornar verd ou falso .	mostre não 2 > 3
-----	-----------	---	------------------

Eis uma lista de comandos relacionados a números:

Primitivas	Modelo	Uso	Exemplos
abs o l u t o abs	absoluto a	Devolver o valor absoluto de a .	abs -23
arredonde	arredonde a	Devolve o inteiro de a .	arredonde 23.4
diferença	diferença a b	Devolve a diferença dos números a e b .	diferença 23 43
exp	exp a	Devolve o valor do número natural (2.7182...) elevado ao expoente a .	exp 2
ln	ln a	Devolve o logaritmo natural de a .	ln 100
log10	log10 a	Devolve o logaritmo de a na base 10.	log10 100
menos	menos a	Troca o sinal de a .	menos 23
pi	nenhum	Devolve o valor 3.141592653589793.	mo pi
potência	potência a b	Devolve a elevado à potência b .	potência 2 3
produto	produto a b	Devolve o produto dos números a e b .	produto 23 43
quociente	quociente a b	Devolve o quociente dos números a e b .	quociente 23 43
raizq	raizq a	Devolve a raiz quadrada de a .	raizq 9
resto	resto a b	Devolve o resto de a e b .	resto 2 3
soma	soma a b	Devolve a soma dos números a e b .	soma 23 43
sorteie	sorteie n	Sorteia um inteiro positivo menor que n .	sorteie 23
acos, arc- cosseno	acos a	Devolve o ângulo cujo cosseno é a .	acos 0.8
asen, arc- seno	asen a	Devolve o ângulo (em graus) cujo seno é a .	asen 0.5
atan, arc- tangente	atan a	Devolve o ângulo cuja tangente é a .	atan 0.8
cos, cosse- no	cos a	Devolve o cosseno de a .	cos 45
sen, seno	sen a	Devolve o seno de a (a em graus).	sen 45
tan, tan- gente	tan a	Devolve a tangente de a .	tan 45

Importante: cuidado com primitivas que requerem dois parâmetros. Por exemplo: em **mudexy a b**, se **b** for negativo, o interpretador Logo fará a operação 200-10 (isto é, ele subtrairá 10 - de 200).

Disso resulta que há somente um parâmetro (190) e não dois, gerando uma mensagem de erro. Para evitar esse tipo de problema, utilize a primitiva "menos" para especificar o número negativo: **mudexy 200 menos 10**.

5.5 OPERAÇÕES COM LISTAS

Primitivas	Parâmetros	Uso	Exemplo
apague	a b	Remove (apaga) o elemento a da lista b .	mo apague 2 [1 2 3 2 4]
conte	a	Retorna o número de elementos de uma lista ou palavra.	mo conte [2 3 oi] mo conte "vila"
elem, elemento	a b	Devolve o "aésimo" elemento de uma lista ou palavra.	mo elem 2 "vila" mo elem 3 [2 4 6]
escolhe	a	Se a é uma palavra, devolve uma de suas letras. Se é uma lista, devolve um de seus elementos.	mo escolhe "vila" mo escolhe [1 2 oi]
inverte	a	Inverte a ordem dos elementos na lista a .	mo inverte [1 2 3]
juntenufim, jf	a b	Coloca a no fim da lista b .	mo jf 3 [7 oi]
juntenuinício, ji	a b	Coloca a no início da lista b .	mo ji 3 [7 oi]
lista	a b	Retorna a lista [a b].	mo lista "oi 3"
palavra	a b	Retorna uma palavra composta pela concatenação de a e b .	mo pal "Xlo "go"
pri, primeiro	a	Retorna o primeiro elemento de uma lista ou palavra.	mo pri [2 3 oi] mo pri "vila"
sentença, sn	a b	Retorna uma lista formada pela concatenação de ab . Os argumentos podem ser palavras ou listas.	mo sn [oi] "oi"
semprimeiro, sp	a	Retorna a lista ou palavra sem o seu primeiro elemento.	mo sp [2 3 4 oi] mo sp "cidadão"

su, semúltimo	a	Retorna a lista ou palavra sem o seu último elemento.	mo su [2 3 4 oi] mo su "cidadão"
ult, último	a	Retorna o último elemento de uma lista ou palavra.	mo ult [2 3 oi] mo ult "vila"

5.6 BOOLEANOS

Um **booleano** é uma primitiva que devolve a palavra “verd ou a palavra falso”. Elas terminam em ponto de interrogação.

Primitivas	Parâmetros	Uso
eant? éanterior?	a b	Retorna verd se a for número (ou letra, ou palavra) anterior a b (em ordem numérica ou alfabética).
éinteiro?	a	Retorna verd se a for um número inteiro; falso, se não.
elista? élista?	a	Retorna verd se a for uma lista.
émembro? emembro?	a b	Retorna verd se a pertencer à lista ou palavra b .
enum? éúmero?	a	Se a for um número, retornará verd. Do contrário, retornará falso .
eprim? éprimitiva?	a	Informa verd se a for uma primitiva do XLogo. mo eprim? "parafrente.
eproc? éprocedimento?	a	Informa verd se a for um procedimento. mo eproc? "quadrado.
éuselápis?	nenhum	Informa verd se o lápis estiver sendo utilizado (a tat desenha ao se mover?).
évariável? evar?	a	Retorna verd se a for uma variável definida.
evazio?évazio?	a	Retorna verd se a for uma palavra ou lista vazia.
falso	nenhum	Retorna falso .
membro	a b	Se b for uma lista (ou palavra), localizará o elemento a na lista. Há duas possibilidades: se a pertencer a b , devolve uma sublista (ou palavra) contendo todos os elementos a partir da posição de a ; se a não pertencer a b , devolve falso. mo membro "a "cidadão mo membro 3 [1 2 3 oi 5].

epal?, épalavra?	a	Se a for uma palavra, retornará verd. Do contrário, retornará falso sãoiguais? a b Retorna verd se a e b forem iguais. Eg.: mo sãoiguais? "c "C mo sãoiguais? "c "c
verd	nenhum	Retorna verd.
visível?	nenhum	Informa se tat aparece (se está visível).

5.7 TESTAR UMA EXPRESSÃO

Como em toda linguagem de programação, o Logo permite verificar se uma condição é satisfeita e então executa o código desejado se ele for verdadeiro ou falso. Com a primitiva **se**, você pode realizar tais testes. Eis a sintaxe:

```
se expressão_a_testar [ lista1 ] [ lista2 ].
```

Se a expressão_a_testar for verdadeira, as instruções da lista1 serão executadas. Por outro lado, se a expressão_a_testar for falsa, as instruções na lista2 serão executadas. A segunda lista é opcional (nesse caso, nada será feito quando a expressão for falsa).

Exemplos:

```
se 1+2=3 [escreva "verdadeiro] [escreva "falso]
se "Y = (primeiro "XLOGO) [pf 100 pd 90] [esc [ XLOGO inicia com um X! ]
se (3*4)=6+6 [mo 12]
```

5.8 LIDAR COM PROCEDIMENTOS E VARIÁVEIS

Procedimentos são como um "programa". Quando um procedimento é chamado, as instruções em seu corpo são executadas. É definido sempre pela palavra-chave "aprenda" e encerrado pela palavra-chave "fim."

```
aprenda nome_do_procedimento :v1 :v2 :v3 ....
```

```
Corpo do procedimentos
```

```
fim
```

nome_do_procedimento é o nome dado ao procedimento;

:v1 :v2 :v3 aguardam por variáveis utilizadas internamente por este procedimento (variáveis locais);

corpo do procedimento representa os comandos a serem executados ao chamarmos o procedimento.

Por exemplo:

```
aprendaquadrado :s
```

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

```
fim
```

O procedimento é chamado **quadrado** e leva um parâmetro chamado **s**. Exemplo: `quadrado 100` desenhará um quadrado square de comprimento 100 (veja mais exemplos de procedimentos no final deste tutorial).

Desde a versão 0.7c, é possível inserir comentários no corpo dos procedimentos desde que precedidos pelo símbolo **#**. Outros programas Logo, tais como Superlogo e Micromundos utilizam o símbolo **;** (ponto-e-vírgula).

Nada do que é escrito à direita do **#** será executado pelo interpretador.

```
aprenda quadrado :s
```

```
#esse procedimento desenha um quadrado de lado igual a :s.
```

```
repita 4[pf :s pd 90] # útil, não?
```

```
fim
```

Importante: não é possível incluir comentários na mesma linha do `aprenda` nem do `fim`.

5.9 RASTREAR

É possível acompanhar o desenrolar de um procedimento. Esse modo também permite ver que os comandos retornam com ajuda da primitiva **saída**. Para ativá-lo, escreva:

```
rastreie verd
```

Naturalmente, **parerastreie** desativará o modo “rastreie”. Um pequeno exemplo com o fatorial:

```

rastreie mo fac 4
fac 4
fac 3
fac 2
fac 1
fac devolva 1
fac devolva 2
fac devolva 6
fac devolva 24
24

```

5.10 ACESSAR ARQUIVOS

Primitivas	Argumentos	Uso
abra	w: palavra	Abre e lê o arquivo w . Por exemplo, para apagar todos os procedimentos definidos a abrir (carregar) o arquivo teste.lgo, você escreveria: eliminetudo abra "teste.lgo . A palavra especificada é um caminho relativo de inicialização do diretório atual. Este comando não funcionará com caminho absoluto.
abrafluxo	id arq	Permite ler ou escrever arquivo que foi chamado arq . O argumento arq deve ser o nome do arquivo desejado. Pode-se utilizar uma lista para indicar o caminho absoluto ou uma palavra para indicar o nome do arquivo no diretório corrente. O argumento id é o número que atribuímos ao fluxo a fim de poder identificá-lo.
carregueimagem carimg	a: palavra	Carrega o arquivo de imagem contida na lista. Seu canto mais elevado esquerdo será colocado onde está a tartaruga. Os formatos suportados são o PNG e o JPG . O caminho especificado deve ser parente ao diretório atual. Exemplo: <code>carimg "tortue.jpg</code> .
dir, diretório	nenhum	Fornece o diretório atual. O padrão é o diretório do usuário, ou seja, /home/seu_login no Linux, C:\WINDOWS no Windows.
escrevalinha eslfx	id lista	Escreve no arquivo indicado por id a linha de texto contida em lista. Atenção, a escrita é efetiva apenas quando fecha-se o fluxo com a primitiva fechefluxo .
fechefluxo	id	Fecha o fluxo identificado pelo número id .

fimarq? fimfluxo?	id	Retorna " verd (verdadeiro) se chegar no fim do arquivo. Caso contrário, retorna " falso .
guardado	w: palavra	" Teste.lgo guarda no arquivo teste.lgo (que já foi salvo no diretório em uso) todos os procedimentos definidos. Se a extensão .lgo for omitida, ela será adicionada automaticamente. A palavra especificada é um caminho relativo de inicialização do diretório atual. Este comando não funcionará com caminho absoluto.
guarde	w: palavra l: lista	Um bom exemplo para explicar isto: guarde " teste.lgo [proc1 proc2 proc3] "; guarda no arquivo teste.lgo no diretório atual os procedimentos proc1, proc2 e proc3. Se a extensão .lgo for omitida, ela será adicionada automaticamente. A palavra especificada é um caminho relativo de inicialização do diretório atual. Este comando não funcionará com caminho absoluto.
juntelfx	id lista	Escreve no fim do arquivo ("junta") a linha de texto contida em lista. Atenção, a escrita é efetiva apenas após fechar o fluxo com a primitiva fechefluxo .
leiacarfx leiacaracterfluxo	id	Abre o fluxo indicado pelo número id e lê um caractere neste arquivo. Esta primitiva retorna um número que representa o valor do caractere (semelhante à leiacar).
leialinha	id	Lê o conteúdo da linha do arquivo identificado por id (lê somente uma linha de cada vez. É necessário repetir o comando para passar para a próxima linha).
listefluxo	nenhum	Lista os diferentes fluxos abertos e seus respectivos identificadores.
marq, mostrearquivos	nenhum	Por padrão, lista o conteúdo do diretório. Equivalente ao comando ls no Linux e o comando dir no DOS.
mudedir, mude- diretório	l: lista	Especifica o diretório a ser utilizado. O caminho deve ser absoluto. O diretório tem que ser especificado dentro de uma lista e o caminho não pode conter espaços.

Eis um exemplo de utilização das primitivas que permitem ler e escrever em arquivos. Apresentaremos esse exemplo para uma arquitetura de tipo Windows. Os outros utilizadores saberão adaptar o exemplo seguinte, pois a ideia é a mesma.

O objetivo é criar o arquivo `c:exemplo.txt` contendo as três linhas:

```
ABCDEFGHIJKLMNOPQRSTUVWXYZ
```

```
abcdefghijklmnopqrstuvwxy
```

```
0123456789
```

```
# Abrimos um fluxo para o arquivo desejado. O fluxo será identificado pelo número 2
```

```
abrafluxo 2 [c:\exemplo.txt]
```

```
# Escrevemos as linhas desejadas
```

```
escrevalinha 2 [ABCDEFGHIJKLMNPOQRSTUVWXYZ]
```

```
escrevalinha 2 [abcdefghijklmnopqrstuvwxy]
```

```
escrevalinha 2 [0123456789]
```

```
# Fechamos o fluxo para que seja registrado ("gravado") o que escrevemos fechefluxo 2
```

Vejamos se a escrita foi bem feita:

```
# Abrimos um fluxo para ler. Dessa vez, o fluxo será identificado pelo número 0 (apenas para mostrar que esse número também pode ser usado);
```

```
abrafluxo 0 [c:\exemplo.txt]
```

```
# Leem-se as linhas dos arquivos sucessivamente:
```

```
mo leialinha 0
```

```
mo leialinha 0
```

```
mo leialinha 0
```

```
# Fechamos o fluxo
```

```
fechefluxo 0
```

Se desejarmos acrescentar a linha "Formidável!":

```
abrafluxo 1 [c:\exemplo.txt]
```

```
juntelfx 1 [Formidável !]
```

```
fechefluxo 1
```

Lembramos mais uma vez que os exemplos acima foram feitos para Windows. No Linux (ou outros sistemas operacionais), a ideia será a mesma. Em vez de [c:\exemplo.txt], teremos algo como [/home/logindousuario/exemplo.txt].

5.11 FUNÇÕES PINTE E PINTEZONA

A tartaruga não deve estar sobre um *pixel* (ponto) da cor da figura que se deseja preencher (se quiser pintar de vermelho, não deverá estar sobre o vermelho).

Vejamos o exemplo para explicar as diferenças entre as duas primitivas: a tat está sobre um ponto de cor branca.

A primitiva **pinte** colorirá todos os *pixels* brancos com a cor do lápis que a tat está utilizando. Se, por exemplo, escrevermos:

```
mudecl 1 pinte
```

Voltemos ao primeiro caso (figura 11). Se a cor do lápis da tat for preta, a primitiva **pintezona** colorirá toda a região até encontrar cores diferentes daquela sobre a qual ela estava (neste exemplo, ao atingir as linhas preta ou verde).

A primitiva **pinte** permite preencher uma área delimitada. Ela pode ser comparada com o “baldinho” disponível em muitos programas de imagens (Paint, Tuxpaint, Gimp, etc.).

Há duas regras a considerar sobre o uso dessa primitiva:

- o lápis deve estar em uso (ul);
- a tat não pode estar sobre um *pixel* com a mesma cor que será utilizada para preencher a área (se desejar pintar algo de vermelho, a tat não pode estar sobre um ponto vermelho).

Eis um bom exemplo:

```
aprenda vai
```

```
dt rainbow 400 ub pe 90 pf 20 pt 120 lp un pd 90 pf 20 ul
```

```
mudecl 0 pinte dep
```

```
mudecl 1 pinte dep
```

```
mudecl 2 pinte dep
```

```
mudecl 3 pinte dep
```

```
mudecl 4 pinte dep
```

```
mudecl 5 pinte dep
```

```
mudecl 6 pinte dep
```

```

fim
aprenda semicirc :c
#desenha um semicírculo com diâmetro :c
repita 180 [pf :c*tan 0.5 pd 1]
pf :c*tan 0.5
pd 90 pf :c
fim
aprenda rainbow :c
se :c<100 [pare]
semicirc :c pd 180 pf 20
pe 90
rainbow :c-40
fim
aprenda dep
un pd 90 pf 20 pe 90 ul
fim
vai

```

6 RECEBER EM ENTRADA DE USUÁRIO

6.1 INTERAGIR PELO TECLADO

No momento, textos do usuário são recebidos por meio de três primitivas.

- **étecla?**: lê verdadeiro (verd) se uma tecla for pressionada ou falso (falso) se não for pressionada;
- **leiacar**: se for falso, o programa é interrompido até que o usuário pressione uma tecla; se **leiacar** for verdadeiro, informa a tecla que foi pressionada por último. Cada tecla possui um valor: se você não estiver certo sobre o valor de uma tecla, você pode digitar: **mo leiacar**. O interpretador esperará que você pressione uma tecla para lhe fornecer o seu valor correspondente.

- **leia lista_título "palavra**: apresenta uma caixa de diálogo com título **lista_título**. O usuário pode então entrar com uma resposta no campo texto, que será armazenada na forma de uma lista na variável **:palavra**, e será avaliada ao clicar em **OK**.

Exemplo:

```
[Gosta de logo?] [Sim ou não] "var
mo :var
aprenda vintage
leia [Qual a sua idade?] "idade
atr "idade pri :idade
se: idade<18 [mo [você é de menor]]
se ou :idade=18 ?idade.18 [mo [você é de maior]]
se :idade>99 [mo [Respeito é bom!]]
fim
```

```
aprenda rali
se étecla? [atr "car leiacar se :car=37 [pe 90]
se :car=39 [pd 90]
se :car=38 [pf 10]
se :car=40 [pt 10]
se :car=27 [pare]]
rali
fim
```

Você pode controlar a tat com as setas do teclado e parar com a tecla Esc

6.2 INTERAGIR COM O MOUSE

No momento, há três primitivas que tratam eventos com *mouse*:

- **leiamouse**: o procedimento é interrompido até que o usuário movimente o *mouse* ou clique em um de seus botões. Então devolverá um número que representa o evento.

Esses eventos são representados pelos valores:

- 0 -> O *mouse* foi movido;
- 1 -> O botão 1 foi pressionado;
- 2 -> O botão 2 foi pressionado, etc.

O botão 1 é o esquerdo; o botão 2, o direito (normalmente, os botões são numerados da esquerda para a direita);

- **posmouse**: devolve uma lista que contém a posição do *mouse*;
- **mouse? Ou rato?**: devolve verdadeiro (verd) ou falso, conforme agiu-se ou não sobre o *mouse* (rato) desde o início da execução do programa.

Neste primeiro exemplo, a *tat* segue o *mouse* quando é movido na tela.

aprenda exemplo

```
# verifica a posição do mouse (leiamouse). A tat se desloca (mudepos) para a posição do mouse (posmouse)
```

```
se leiamouse = 0 [mudepos posmouse]
```

```
exemplo
```

```
fim
```

Neste segundo exemplo, você deverá clicar com o botão esquerdo do *mouse* para que a *tat* se mova.

aprenda exemplo2

```
se leiamouse = 1 [mudepos posmouse]
```

```
exemplo2
```

```
fim
```

No terceiro exemplo, criamos dois botões (em cor-de-rosa). Ao clicar com o botão esquerdo do *mouse*, desenha-se um quadrado de lado 40. Ao clicar com o botão esquerdo do *mouse* no botão da direita, desenha-se um pequeno círculo.

Por fim, ao clicar com o botão direito do *mouse* no botão da direita, o procedimento será interrompido.

aprenda vai

```
limpedesenho botão un mudepos [150 0] ul botão
```

```
un mudepos [22 20] ul rotule "quadrado
un mudepos [180 20] ul rotule "Círculo
un mudepos [0 -100] ul
mouse
fim
```

```
aprenda botão
#cria botão retangular cor-de-rosa (altura 50: largura 100)
repita 2[pf 50 pd 90 pf 100 pd 90]pd 45 un pf 10 ul mudecl [255153 153]
pinte pf 10 pe 45 ul mudecl 0
fim
```

```
aprenda mouse
# o valor de leia mouse é armazenado na variável ev
atr "ev leia mouse
# o primeiro valor da coordenada da posição do mouse é armazenada na
variável x
atr "x elem 1 posmouse
# o segundo valor da coordenada da posição do mouse é:
armazenada na variável y
atr "y elem 2 posmouse
# O que acontece ao clicar com o botão esquerdo do mouse:
se :ev=1 & :x>0 & :x<100 & ?y>0 & :y<50 [quadrado]
# O que acontece ao clicar com o botão direito do mouse:
se :x>150 & :x<250 & ?y<50 [
se :ev=1 [circule]
se :ev=3 [pare]]
mouse
fim
```

```
aprenda circule
```

```
repita 90 [pf 1 pe 4] pe 90 un pf
```

```
40 pd 90 ul
```

```
fim
```

```
aprenda quadrado
```

```
repita4 [pf 40 pd 90] pd 90 pf 40
```

```
pe 90
```

```
fim
```

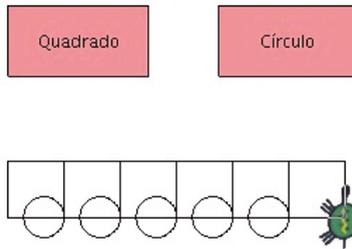


Figura 6 – Uso do mouse

6.3 INTERAGIR POR IG (INTERFACE GRÁFICA)

O XLogo possibilita a criação de alguns componentes gráficos na área de desenho (botão, menu, etc.). Esses componentes estão associados às interfaces gráficas e, para facilitar todas essas primitivas, iniciam por “ig” (interface gráfica). São as primitivas **igbotão**, **igmenu** e **iglocal**, e suas complementares **igfaz**, **igapague** e **igdesenhe**.

Para criar um componente: para utilizar esses objetos gráficos, é necessário criá-los e definir alguns parâmetros. Para criar um botão, utilizamos a primitiva **igbotão**.

```
# Este comando cria um botão identificado pelo nome b
```

```
# No botão aparecerá escrito: Clique
```

```
igbotão “b “Clique
```

Para criar um menu de rolagem, utilizamos a primitiva **igmenu**.

```
# Este comando cria um menu de rolagem identificado pelo nome m
```

```
# e conterà 3 entradas: item 1, item 2 e item 3
```

```
igmenu " m [item1 item2 item3]
```

Como atribuir propriedade a estes componentes?

- **Iglocal:** permite posicionar (localizar) o elemento gráfico onde desejar na área de desenho. Por exemplo, para posicionar o botão definido acima no ponto (20, 100), escrevemos:

```
iglocal "b [20 100]
```

Se o local do componente não for indicado, ele será localizado, por padrão, no canto superior esquerdo da área de desenho.

- **Igapague:** apaga (suprime, delta) um elemento gráfico. Por exemplo, para apagar o botão acima:

```
igapague "b
```

- **igfaz:** define o que o elemento gráfico faz quando o usuário clicar sobre ele.

```
# A tat avançará 100 passos se clicarmos sobre o botão "b
```

```
igfaz " b [pf 100]
```

```
# Para o menu de rolagem, cada item contém sua própria ação
```

```
igfaz " m [[mostre "item1] [mostre "item2] [mostre "item3]]
```

- **igdesenhe:** mostra o componente gráfico na área de desenho. É importante, pois no Xlogo, o botão ou menu só será exibido com esse comando. Por exemplo, para vermos o botão **b**:

```
igdesenhe "b
```

Como exemplo, vamos reescrever os procedimentos criados em interações com o *mouse*. Note que, desta vez, o projeto fica bem mais econômico, por assim dizer.

```
aprenda vai
```

```
limpedesenho botão
```

```
un mudepos [0 -100] ul
```

```
fim
```

aprenda circule

repita 90 [pf 1 pe 4] pe 90 un pf 40 pd 90 ul

fim

aprenda quadrado

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

fim

aprenda botão

#cria a botão "bQ" com o rótulo "Quadrado" igbotão "bQ "Quadrado

#Posiciona o botão "bQ iglocal "bQ [-80 100]

#Informa ao botão que ele executará o procedimento quadrado ao ser clicado

igfaz "bQ [quadrado]

#Exibe o botão b"

igdesenhe "bQ

#A seguir, a mesma ideia geral para o botão "bC" que desenhará um círculo:

igbotão "bC Círculo

iglocal "bC [80 100]

igfaz "bC [circuloe]

igdesenhe "bC

fim

7 TEMPO E DATA

O XLogo possui muitas primitivas para data, tempo ou contadores.

espere	n: um número	Para o programa por n/60 segundos.
contador	n: inteiro	Inicia um contador de n segundos. Sabemos se ele terminou por meio da primitiva fimcontador? . As primitivas contador e fimcontador? funcionam como um cronômetro.
fimcontador?	nenhum	Devolve " verd " se não há contador ativo. Devolve " falso " se o contador está ativo.
data	nenhum	Devolve uma lista contendo três inteiros a representar a data. O primeiro inteiro indica o dia do mês; o segundo, o mês; o terceiro, o ano --> [dia mês ano]
tempo	nenhum	Devolve uma lista de inteiros no formato --> [hora minuto segundo]
tempodec tempodecorrido	nenhum	Devolve o tempo decorrido (em segundos) desde que o XLogo foi iniciado.

A diferença entre **espere** e **contador** é que o **contador** não interrompe o programa. Um exemplo:

```
# Exibe o tempo no formato numérico
# Atualiza a cada 5 segundos
se fimcontador? [
ld
mudeonte 75 dt
atr "hora tempo
atr "h pri :hora
atr "m elem 2 :hora
# Exibir dois algarismos para segundos e minutos. (adiciona um 0)
se :m-10<0 [atr "m pal 0 :m]
atr "s ult :hora
# Exibir dois algarismos para segundos e minutos. (adiciona um 0)
se :s-10<0 [atr "s pal 0 :s]
rotule pal pal pal pal :h " : :m " : :s
contador 5 ]
relogio
fim
```

8 EXEMPLOS DE PROGRAMAS

8.1 DESENHAR CASAS

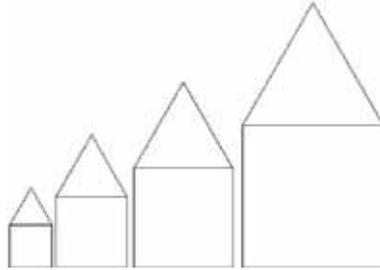


Figura 7 – Casas

```

aprenda casa :c
repita 4[pf 20*:c pd 90]
pf 20*:c pd 30
repita 3[pf 20*:c pd 120]
fim
  
```

```

aprenda place :c
un pe 30 pt :c*20 pd 90 pf :c*22 pe 90 ul
fim
  
```

```

aprenda hut
limpedesenho un pe 90 pf 200 pd 90 ul dt
casa 3 place 3 casa 5 place 5 casa 7 place
7 casa 10
fim
  
```

8.2 DESENHAR UM RETÂNGULO PREENCHIDO



Figura 8 – Retângulo

```

aprenda rect :lo :la
se :lo=0 | :la=0 [pare]
repita 2[pf :lo pd 90 pf :la pd 90]
rect :lo -1 :la -1
fim
  
```

O símbolo | na segunda linha equivale à primitiva "ou."

```
aprenda rect :lo :la
se ou :lo=0 :la=0 [pare]
repita 2[pf :lo pd 90 pf :la pd 90]
rect :lo -1 :la -1
fim
```

Uma forma mais simples (talvez menos elegante) seria:

```
aprenda rect1 :lo :la
repita 2[pf :lo pd 90 pf :la pd 90]
un pd 45 pf 1 ul pinte
fim
```

8.3 FATORIAL

Lembre-se de que:

$$5! = 5 \times 4 \times 3 \times 2 \times 1 = 120$$

```
aprenda fac :n
se :n=1[saída 1][saída :n*fac :n-1]
fim
mostre fac 5
120.0
mo fac 6
720.0
```

8.4 FLOCO DE NEVE²

```
aprenda koch :ordem :len
se :ordem < 1 | :len < 1
[parafrente :len pare]
koch :ordem-1 :len/3
pe 60
koch :ordem-1 :len/3
pd 120
koch :ordem-1 :len/3
pe 60
koch :ordem-1 :len/3
fim
aprenda flocokoch :ordem :len
repita 3 [pd 120 koch :ordem
:len]
fim
flocokoch 5 450
```

² Agradecimento a Georges Noël.

aprenda lindarosacea
rosacea
repita 30 [pf 2 pd 1]
rosacea
repita 15 [pf 2 pd 1]
rosacea
repita 30 [pf 2 pd 1]
rosacea
fim
ul ld mudecf 0 mudedl 5 dt rosacea un mudepos [-300 0] ul mudedireção
0 lindarosacea

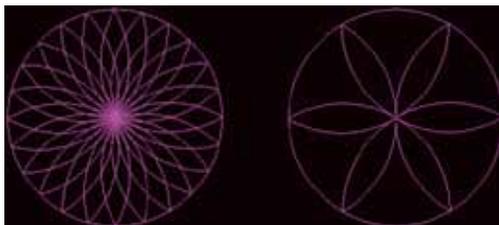


Figura 10 – Melhor do que utilizar um compasso

9 INSTALAR, DESINSTALAR E ATUALIZAR

a) Instalar – para instalar o XLogo, é necessário que você tenha instalado em seu computador o “Java Runtime Environment” (JRE) produzido pela Sun.

Importante: esse programa pode ser copiado gratuitamente em java.sun.com/products/archive/j2se/1.4.2/index.htm ou em java.sun.com/j2se/1.5.0/download.jsp.

No Windows, uma forma de você saber se tem o Java instalado e em funcionamento é: no menu **Iniciar**, clique em **Executar**, escreva **javaw** e dê **OK**. Se nenhuma mensagem de erro aparecer é porque está tudo certo.

Para saber mais detalhes sobre como instalar o XLogo em Linux ou Windows, leia as instruções em br.geocities.com/projetologo/logo/slogo.html.

b) Desinstalar – para desinstalar o SL, basta deletar o arquivo **xlogo.jar** (ou **xlogo-new.jar**) e o arquivo de configuração **.xlogo** que está localizado no diretório **home** (**/home/seu_login**) para usuários Linux na mesma pasta do arquivo **xlogo.jar** (ou **xlogo-new.jar**) ou ainda **c:\windows\xlogo**.

c) Atualizar – confira a última versão e correção de *bugs* no sítio do XLogo com frequência (<http://xlogo.tuxfamily.org>).

Sinta-se à vontade para contatar o autor do programa quanto à instalação ou uso (inglês ou francês). Todas as sugestões são bem-vindas.

Para suporte em português, acesse <<http://xlogo.tuxfamily.org/pt>>.

10 LISTA DE PRIMITIVAS

Português	Francês	Espanhol	Inglês	Alemão
abra	ramene	carga	load	lade
abrafluxo	ouvreflux	abreflujo	openflow	oeffnefluss
absoluto, abs	abs	absoluto	abs	ovz
adeus	bye aurevoir	adios	bye	ade
animado	animation	animacion	anim	animation
apague	enleve	quita	remove	entf
aprenda	pour	para	to	lerne
arco	arc	arco	arc	arc
arredonde	arrondi	redondea	rnd	rnd
atênçãotat, atat	fixetortue, ftortue	pontortuga, ptortuga	setturtle, sturtle	setzeschildkroete, ssk
atribua, atr	donne	haz	globalmake, make	setze
atrlocal	donnelocale	hazlocal	localmake	lokalsetze
baixenariz	pique	cabeceaabajo	downpitch	runter
carregueimagem, carimg	chargeimage, ci	cargaimagen, ci	loadimage, li	ladebild, lb
centro	origine	centro	home	heim
cercar	clos	modojaula	close	schl
chattcp	chattcp	chattcp	chattcp	chattcp
círculo	cercle	circulo	circle	kr
comlímite	enr	modovuelta	wrap	rspr
contador	debuttemps	crono	countdown	zr

conte	compte	cuenta	count	zaehle
conteúdo	chose	objeto	thing	wert
convezes, cv	compteur	cuentarepite, contador	repcount	wiederholungs- zahl
cordofundo, cdf	couleurfond, cf	colorpapel	screencolor, sc	bildfarbe, bf
cordolápis, cdl	couleurcrayon, cc	colorlapiz, cl	pencolor, pc	stiftfarbe, sf
cordoponto, cdp	trouvecouleur, tc	encuentracolor, ec	findcolor, fc	findefarbe, ff
cordotexto	couleurtexte	colortexto	textcolor	farbtext
data	date	fecha	date	datum
deletesequên- cia, delseq	effacesequence, efseq	borrasecuencia, bos	deletesequence, delseq	loeschefolge, lschfol
defina, def	def, definis	def, define	def, define	def, definiere
diferença	difference	diferencia	diff	diff
direção, dç	cap	rumbo	heading	kurs
direção para, dç para	vers	hacia	towards	richtung, ri
diretório, dir	repertoire, rep	directorio, dir	directory, dir	ordner, ord
distância, dist	distance	distancia	distance	abstand, abst
e	et	y	and	und
é anterior?, eant?	precede?	antes?, ante- rior?	before?, beforep	vorher?
é inteiro?	entier?	entero?	integer?	ganzzahl?
edite	ed	ed	ed	ed
editetudo	edtout	edtodo	edall	edalles
eixo	axes	ejes	axis	achsen
eiox	axex	ejex	xaxis	xachse
eioy	axey	ejey	yaxis	yachse
eixo?	axes?	ejes?	axis?	achsen?
eiox?	axex?	ejex?	xaxis?	xachse?
eioy?	axey?	ejey?	yaxis?	yachse?
elemento, elem	item	elemento	item	element
elimine	effacenom	borra	erase	vergiss

eliminetudo, et	effacenoms, efns	borratodo, bot	erall	vergissalles, vga
élista?, elista?	liste?	lista?	list?, listp	liste?
émembro?, emembro?	membre?	miembro?	member?, memberp	element?
énúmero?, enum?	nombre?	numero?	numberp, num- ber?	zahl?
enquanto	tantque	mientras	while	solange
enviatcp	envioietcp	enviatcp	sendtcp	sendetcp
épalavra?, epal?	mot?	palabra?	word?, wordp	wort?
éprimitiva?, eprim?	primitive?, prim?	primitiva?, prim?	primitive?, prim?	grundwort?, gw?
éprocedimen- to?, eproc?	procedure?, proc?	procedimiento?, proc?	procedure?, proc?	prozedur?, proz?
escolhe	choix	elige	pick	nimmwas
escondetat, dt	cachetortue, ct	ocultatortuga, ot	hideturtle, ht	versteckeschil- dkroete
escreva, esc	tape	tipea	write	schreibe
escrevalinha, eslfx	ecrisligneflux	escribelineaflujo	writelineflow	schreibezeil- enfluss
escutetcp	ecoutetcp	escuchatcp	listentcp	wartetcp
espere	aattends	espera	wait	warte
espezzuradolá- pis, el	taillecrayon, tc	grosorlapiz, gl	penwidth, pw	
étecla?	touche?	tecla?	key?	taste?
éuselápis?	baisseccrayon?	bajalapiz?	pendown?	stiftab?
execute	exec	ejecuta	run	starte
executetcp	executetcp	ejecutatcp	executetcp	ausfuehrentcp
évazio?, evazio?	vide?	vacio?	empty, empty?	leer?
exp	exp	exp	exp	exp
falso	faux	falso	false	falsch
fechefluxo	fermefflux	cierraflujo	closeflow	schliessefluss?
fim	fin	fin	end	ende
fimarq?, fimfluxo?	finflux?	finflujo?	endflow?	endefluss?

fimcontador?	fintemps?	fincrono?	endcountdown?	ezr?
fonte	taillepolice	fuelle	fontsize	schrf
fontetexto, ftexto	tpt, taillepolice	fuentetexto, ftexto	textsize, ts	schriftgroesse
grade	grille	cuadrícula	grid	gitter
grade?	grille?	cuadrícula?	grid?	gitter?
guardado	sauved	guardatodo	saved	gespeichert
garde	sauve	guarda	save	speicher
igapague	guienleve	eliminaigu	guiremove	entfernegui
igbotão	guibouton	botonigu	guibutton	guitaste
igdesenhe	guidessine	dibujagu	guidraw	zeichnegui
igfaz	guiaction	accionigu	guiaction	guiaktion
iglocal	guiposition	posicionigu	guilocation	guiposition
igmenu	guimenu	menuigu	guimenu	guimenu
imts	imts	imts	poall	zga
índice seqüência, indseq	indexsequence, indseq	índice secuencia, indsec	indexsequence, indseq	indexfolge, indfol
inicielinha	lignedef	empieza línea	linestart	linienanfang
iniciepolígono	polydef	empieza polígono	polystart	vieleckanfang
inicieponto	pointdef	empieza punto	pointstart	punkteanfang
instrumento, instr	instrument, instr	instrumento, instr	instrument, instr	instrument, instr
inverte	inverse	invierte	reverse	umdrehenliste
invertelápis, il	inversecrayon, ic	inviertelapiz, ila	penreverse, px	umkehrstift, us
juntelfx	ajouteligneflux	agregalineaflujo	appendlineflow	haengezeileanfluss
juntenofim, jf	metsdernier, md	ponultimo, pu	lput	mitletzem, ml
juntenoinício, ji	metspremier, mp	ponprimero, pp	fput	miterstem, me
leia	lis	leeteclado	read	lies
leiacar	liscar	leecar	readchar	lz
leiacarfx	liscarflux	leecarflujo	readcharflow	lesebuchstabenfluss

leialinha	lisligneflux	leelineaflujo	readlineflow	lesezeilefluss
leiamouse	lissouris	leeraton	readmouse	liesmaus
levantenariz	cabre	cabeceaarriba	uppitch	rauf
limpedesenho, ld	videecran, ve	borrápantalla, bp	clearscreen, cs	loeschebild, lschb
limpetexto, lt	videtexte, vt	borratexto, bt	cleartext, ct	loeschetext, lt
lista	liste	lista	list	liste
listefluxo	listeflux	listaflujos	listflow	listefluss
listepar	lvars	imvars	variables	lvarn
ln	log	ln	log	log
local	locale	local	local	lokal
log10	log10	log10	log10	log10
mate	efv	bov	kill	vgv
máximodetartarugas, maxtat	maxtortues	maximastortugas	turtlesmax	schildkrötenmax
membro	membre	miembro	member	ela
menos	moins	cs	minus	minus
mensagem, msg	message, msg	mensaje, msj	message, msg	nachricht, nrcht
mostre, mo	ecris, ec	escribe, es	print, pr	druckzeile, dz
mostrearquivos, marq	catalogue, cat	catalogo, cat	files	zeigedateien, zd
mostretat, at	montretortue, mt	muestratortuga, mt	showturtle, st	zeigeschildkroete, zs
mudecordofundo, mudecf	fixecouleurfond- graphique, fcfg	poncolorpapel, poncp	setscreencolor, setsc	setzebildfarbe, sbf
mudecordolápis, mudecl	fixecouleur- crayon, fcc	poncolorlapiz, poncl	setpencolor, setpc	setzestiftfarbe, ssf
mudecordotexto, mudect	fixecouleurtext- e, fct	poncolortexto, pctexto	setttextcolor, stc	setzeschriftfarbe, sschfff
mudedireção, mudedç	fixecap	ponrumbo, ponr	setheading	aufkurs, ak
mudediretório, mudedir	fixerepertoire, cd	pondiretorio, cambiadirec- torio	setdirectory, changedirectory	setzeordner, wechsleordner

mudeespessura-dolápis, mudeel	fixetaillecrayon, ftc	pongrosor	setpenwidth, setpw	stiftdicke, sd
mudefonte, mudef	ftp, fixetaillepolice	ponfuenta, pf	setfontsize, setfs	setzeschriftart, sschrfa
mudefontetexto, mudeft	ftpt, fixetaillepolice-texte	ponfuentetexto, pft	settextsize, sts	setzeschriftart-text, sschrfat
mudeíndice-seqüência, mudeindseq	fixeindexse- quence, findseq	ponindice- cuencia, pindsec	setixeindexse- quence, sindseq	findefolge, findfol
mudeinstru- mento, mudeinstr	fixeinstrument, finstr	poninstrumento, pinstr	setinstrument, sinstr	setzeinstru- ment, sinstr
mudemáximo- detartarugas, mudemaxtat	fixemaxtortues, fmt	ponmaximastor- tugas, pmt	setturtlemax, stm	setzeschil- dkrötenmax
mudenome- fonte, mudenf	fixenompolice, fnp	ponnombre- fuenta, pnf	setfontname, setfn	setzeschriftna- me, sschrfa
mudenomefon- tetexto, mudenft	fixenompolice- texte, fnpt	ponnombre- fuentetexto, pnft	settextname, stn	setzeschrif- tnametext, sschrfnat
mudepontado- lápis, mudepl	fixeformecrayon	ponformalapiz	setpenshape	setzestiftform
mudepos	fpos	ponpos	setpos	spos
mudequalida- dedaimagem, mudeqi	fixequalite- dessin	poncalidaddi- bujo	setdrawing- quality	szeichnung- qualität
muderoupa	fforme	pforma	setshape	sform
mudetamanho- dajanela	fixetailledessin	pontamañopan- talla	setscreensize	
mudex	fixex	ponx	setx	sx
mudexy	fixexy	ponxy	setxy	sxy
mudey	fixey	pony	sety	sy
não	non	no	not	nicht
nomefonte	nompolice, np	nombrefuen- te, nf	fontname	schrfa

nomefontetexto, nft	nompolicetexte, npt	nombrefuente-texto, nft	textname, tn	schriftartname, schrfana
ou	ou	o	or	oder
palavra, pal	mot	palabra	word	wort
para	repetepour	repitepara	for	fuer
paradireita, pd	tournedroite, td	giraderecha, gd	right, rt	rechts, re
paraesquerda, pe	tournegauche, tg	giraizquierda, gi	left, lt	links, li
parafrente, pf	avance, av	avanza, av	forward, fd	vorwaerts, vw
paratrás, pt	recule, re	retrocede, re	back, bk	rueckwaerts, rw
pare	stop	alto	stop	stop
pareanimado	stopanimation	detieneanimacion	stopanim	stoppeanimation
paretudo	stoptout	detienetodo	stopall	hoerauf
perspectiva	perspective	perspectiva	3d	perspektivisch
pi	pi	pi	pi	pi
pinte	remplis	rellena	fill	fuelle
pintezona	rempliszone	rellenazona	fillzone	formfuellen
ponto	point	punto	dot	punkt
pos	pos	pos	pos	pos
posmouse	possouris	posraton	posmouse	posmaus
potência	puissance	potencia	power	pot
primeiro, pri	premier, prem	primero, pr	first erstes	
produto	produit	producto	product	produkt
qualidadedaimagem	qualitedessin	calidaddibujo	drawingquality	zeichnungsqualität
quociente	quotient	cociente	quotient	quotient
raizq	racine, rac	raizcuadrada, rc	squareroot, sqrt	wurzel, wrzl
rastreie	trace	trazado	trace	verfolge
repita	repete	repite	repeat	wh
resto	reste	resto	remainder	rest
roleparadireita, rolepd	roulisdroite	balanceaderecha	rightroll	rollerechts

roleparaesquerda, rolepe	roulisgauche	balanceaiz- quierda	leftroll	rollelinks
rotule	etiquette	rotula	label	sktext
roupa	forme	forma	shape	form
saída	ret	dev	op	rg
sãoiguais?	egal?	iguales?	equalp	gleich?
se	si	si	if	wenn
semeixo	stopaxes	detieneejes	stopaxis	stoppeachsen
semgrade	stopgrille	detienecuadri- cula	stopgrid	stoppegitter
sem limite	fen	modoventana	window	fen
sem primeiro, sp	saufpremier, sp	menos primero, mp	butfirst, bf	ohneerstes, oe
semtartaruga, semtat	tuetortue	eliminatortuga	killturtle	loescheschild- kroete
sem último, su	saufdernier, sd	menosultimo, mu	butlast, bl	ohneletztes, ol
sentença, sn	phrase, ph	frase, fr	sentence, se	satz
seqüência, seq	sequence, seq	secuencia, sec	sequence, seq	olge, fol
soma	somme	suma	sum	plus
sorteie	hasard	azar	ran	zuf
tamanhodajane- la, tamjan	tailledessin	tamaño pantalla, tpant	screen size bildgröße,	bildgrösse
tamanhojane- la, tj	taillefenetre, tf	tamañoventana, tv	zonesize .	
tartarugas	tortues	tortugas	turtles	skn
tatativa	tortue	tortuga	turtle	sk
tempo	heure	hora	time	zeit
tempodecorrido, tempodec	temps	tiempo	pasttime	vergangenzeit, vz
terminelinha	lignefin	finlínea	lineend	linienende
terminepolí- gono	polyfin	finpolígono	polyend	vieleckende
termineponto	pointfin	finpunto	pointend	punkteende
toque	joue	tocamusica	play	spiele

último, ult	der, dernier	ultimo	last	letztes
useborracha, ub	gomme, go	goma, go	penerase, pe	radiere, rd
uselápiz, ul	baissecrayon, bc	bajalapiz, bl	pendown, pd	stiftab, sa
usenada, un	levecrayon, lc	subelapiz, sl	penup, pu	stifthoch, sh
veranimado	rafraichis, rf	refrescar	refresh	neumalen
verd	vrai	cierto	true	wahr
visível?	visible?	visible?	visible?	sichtbar?
zoom	zoom	zoom	zoom	zoom

11 TRIGONOMETRIA

acos, arccoseno	acos, arccosinus	acos, arcocoseno	acos, arccosine	acos, acosinus
asen, arcseno	asin, arcsinus	asen, arcoseno	asin, arcsine	asin, asinus
atan, arctangente	atan	atan	atan	atan
cos, coseno	cos, cosinus	cos	cos, cosine	cos, cosinus
sen, seno	sin, sinus	sen	sin, sine	sin, sinus
tan, tangente	tan, tangente	tan tangente	tan, tangent	tan, tangenz

REFERÊNCIAS

Creative Commons. Projeto Logo 2010: XLogo. Disponível em: <<http://projetologo.webs.com/texto1.html>>. Acesso em: 02 mar. 2009.

SECRETARIA DE ESTADO DA EDUCAÇÃO
Diretoria de Tecnologia Educacional
Rua Salvador Ferrante, 1.651 – Boqueirão
CEP 81670-390 – Curitiba – PR
www.diaadiaeducacao.pr.gov.br