



- A Interface:
 - [A janela principal](#)
 - [O editor de procedimentos](#)

Os Menus:

- [Menu "Arquivo"](#)
- [Menu "Edição"](#)
- [Menu "Ferramentas"](#)
- [Menu "Ajuda"](#)

Convenções Adotadas no XLOGO

- [Comandos e a interpretação](#)
- [Procedimentos](#)
- [O caracter especial «\»](#)
- [Caixa alta](#)
- [Operadores e sintaxe](#)
- [As cores](#)

Uso de Primitivas

- [Mover Tat; configurar fonte, lápis e cor](#)
- [Tat no Espaço](#)
- [Primitivas mostre e escreva](#)
- [Operações aritméticas e de lógica](#)
- [Operações com listas](#)
- [Booleanos](#)
- [Testar uma Expressão](#)
- [Lidar com procedimentos e variáveis](#)
- [Acessar arquivos](#)
- [Funções pinte e pintezone](#)
 - [PreenchaPolígono](#)
- [Comandos de parada](#)
- [Loops \(Laços\)](#)
- [Modo Tartarugas Múltiplas](#)
- [Tocar música](#)
- [Receber entrada \(input\) do usuário](#)
 - Interação pelo teclado
 - Interação pelo mouse (rato)
 - Interação por IG

- [Tempo e data](#)
- [Uso de Rede](#)
- [Exemplos de programas](#)
 - Desenhar casas
 - Desenhar um retângulo preenchido
 - Fatorial
 - O floco de neve
 - Um pouco de escrita...
 - ...E conjugação
 - Primeira versão
 - Segunda tentativa
 - Ou ainda: Um pequena recursão !
 - Tudo sobre cores
 - Introdução
 - Vamos praticar!
 - E se você desejar um negativo??
 - Um bom exemplo de uso de listas
 - Uma linda rosácea

[Instalar/Desinstalar e Atualizar](#)

[Lista de Primitivas](#)

[Sobre esse documento ...](#)



Última atualização: 28/fev/2010

Interface

Subseções

- A janela principal
- O editor de procedimentos

Janela Principal

No topo encontra-se os menus de costume: Arquivo Edição Ferramentas e Ajuda Logo abaixo aparece a linha de comando, onde escreve-se os comandos (instruções). Para a instrução ser executada basta dar "enter". No meio temos a área de desenho. Na parte de baixo encontra-se o histórico de comandos. Ele mostra os comandos já executados bem como as respectivas respostas. Eventuais mensagens de erro aparecem em vermelho. Para chamar um comando já executado basta clicar nele no histórico de comandos, ou usar as setas de teclado (para baixo e para cima) na linha de comando. No canto direito inferior há dois botões: Editor e Pare. PARE interrompe a execução do programa e EDITOR abre a janela do [editor de procedimentos](#).

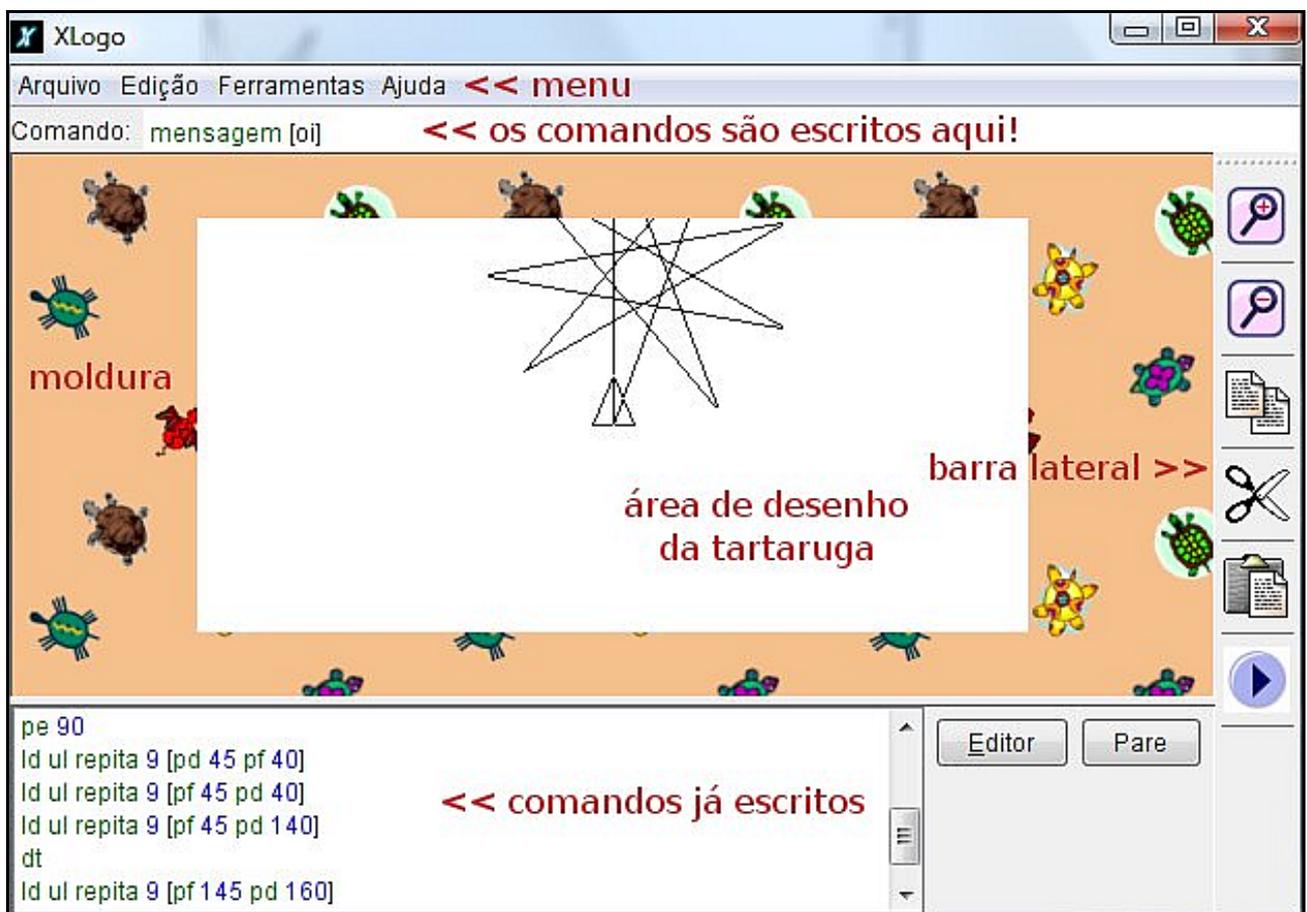
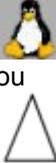


Figura 1: A janela principal do xLogo

Há três modos de abrir o editor:

- Escrever 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 ...
- Pressionar o botão Editor no canto inferior direito da janela do XLOGO.
- Usar as teclas de atalho: Alt+E

Esses são os botões que você encontrará no Editor:

	<p>Fecha o editor e guarda (salva) o que foi feito. Esse é o botão que deve ser pressionado sempre que escrever novos comandos ou fizer alterações nos procedimentos. Se preferir, use as teclas de atalho ALT+Q. Nas versões mais recentes do xLogo, a figura mostrada é a roupa da tat em uso.</p>		<p>Esse 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. Esse comando principal será salvo junto com o arquivo .lgo.</p>
	<p>Sai do editor sem salvar nenhuma das mudanças feitas. Você também pode usar as teclas de atalho ALT+C.</p>		<p>Recorta o texto selecionado para o clipboard</p>
	<p>Imprime o conteúdo do editor.</p>		<p>Cola o texto armazenado no clipboard</p>
	<p>Copia o texto selecionado para o clipboard</p>		<p>Localiza e substitui termos no editor</p>

IMPORTANTE:

- Note que clicar no botão de fechar (x) no canto direito superior (janela de título) não fará efeito algum! Somente os dois primeiros botões citados acima é que permitem sair do editor.
- Para apagar (eliminar) procedimentos, use as primitivas elimine e eliminetudo (et).

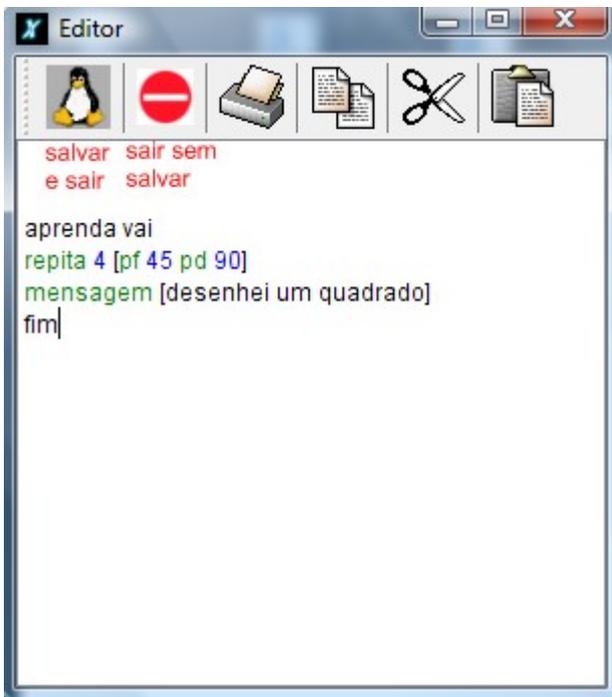


Figura 2: A janela do Editor de Procedimentos

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

Menus

Subseções

- [Menu "Arquivo"](#)
- [Menu "Edição"](#)
- [Menu "Ferramentas"](#)
- [Menu "Ajuda"](#)

Menu Arquivo

- Arquivo -> Abrir: para abrir (carregar) algum arquivo logo já existente.
- Arquivo -> Guardar: guarda os procedimentos do arquivo em uso.
- Arquivo -> Guardar como ...: Pode ser usado para guardar (salvar) os procedimentos atuais em um arquivo com outro nome.
- Arquivo -> Sair: sai (fecha) o xLogo.
- Arquivo -> Capturar imagem -> Guardar imagem como... : permite guardar o desenho exibido na tela do XLOGO como uma imagem nos formatos jpg ou png. Se desejar gravar somente uma parte do desenho, use o mouse conforme as instruções que aparecerão numa caixa de diálogo (um clique no canto superior esquerdo, outro clique 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 acima.

Menu Edição

- Edição -> Copiar: copia o texto selecionado para o clipboard.
- Edição -> Recortar: recorta o texto selecionado para o clipboard.
- Edição -> Colar: cola na linha de comando o texto armazenado no clipboard.

Menu Ferramentas

- Ferramentas -> Mudar cor do lápis: escolha uma cor para o lápis da tartaruga. Também é possível fazê-lo com o comando mudecl.
- Ferramentas -> Mudar cor do fundo: escolha uma cor para o fundo da tela. Também é possível fazê-lo com o comando mudecf.
- Ferramentas -> Arquivo inicial: Informe o caminho para os arquivos de inicialização. Qualquer procedimento contido em arquivos no formato *.lgo se tornarão "pseudo-primitivas" 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: Traduza automaticamente qualquer procedimento de um idioma para outro. Pode ser útil, por exemplo, ao copiar na internet algum projeto em outro idioma.
- Ferramentas -> Apaga procedimentos: Escolha que procedimentos deseja eliminar. Veja a primitiva [elimine](#).
- Ferramentas -> Preferências: Abre uma caixa de diálogo na qual você pode configurar várias coisas:
- Idioma : escolha entre Francês, Espanhol, Inglês ou Português. Note que as primitivas são diferentes em cada idioma.
 - Aparência: Escolha 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 usando a barra deslizante.
 - Na segunda aba, você pode escolher uma "roupa" para a Tat (uma figura para a tartaruga). Nas abas seguintes, pode-se controlar número de tartarugas, estilo do lápis, instrumentos de som (mudeinstrumento número) e fonte (tipo e tamanho).

Menu Ajuda

- Menu -> Licença: exhibe a licença pública geral (GPL) sob a qual esse programa é distribuído.
- Menu -> Tradução da Licença: exhibe uma tradução da licença acima. Essa tradução não é oficial (apenas a versão em inglês o é) e está aqui apenas para ajudar no entendimento.
- Menu -> Traduza o xLogo: Ajude a traduzir o xLogo. O arquivo gerado deve ser enviado para nós.
- Menu -> Sobre...: O de sempree o endereço <http://xlogo.tuxfamily.org> para o marcador do seu navegador (bookmarks) !! o:)

Convenções Adotadas

Essa seção aborda alguns pontos essenciais da linguagem LOGO, e também do xLogo.

Subseções

- [Comandos e a interpretação](#)
- [Procedimentos](#)
- [O caracter Especial "\"](#)
- [Caixa alta](#)
- [Operadores e sintaxe](#)
- [Uso de Cores](#)

Comandos e sua Interpretação

A linguagem LOGO permite que certos eventos sejam disparado por comandos internos - esses comandos são chamados 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.

Os argumentos no LOGO são de três tipos:

- Números: algumas primitivas exigem números como argumento: `pf 100` é um exemplo.
- Palavras: São marcadas pela inicial `"`. Um exemplo de primitiva com argumento é a `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).

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 usando 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 desenha um quadrado.

Os procedimentos também podem trabalhar com argumentos. Para isso, usamos 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 nesse manual.

O Caracter Especial "\"

O caracter «\» (barra invertida ou "backslash") permite em particular colocar espaço entre palavras ou ainda uma nova linha. «\n» faz a quebra de linha enquanto que «\ » seguido de um espaço indica espaço numa palavra. Exemplo:

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

Se desejar escrever o caracter «\» 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 colocar um caractere «\» antes.

```
mo "(xlogo\
  (xlogo)
```

Todos os caracteres «\» são ignorados. Este aspecto é especialmente importante em particular para o gerenciamento de arquivos.

Exemplo no Windows: Para definir (mudar) o diretório atual para C:\Meus Documentos, escreva:

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

Exemplo 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).

Caixa Alta

XLOGO não faz distinção entre maiúsculas e minúsculas para procedimentos e primitivas. Assim, no procedimento quadrado, definido anteriormente, tanto faz escrever QUADRADO ou qUADdRAdO, 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)
```

Operadores e Sintaxe

Há dois modos de escrever certos comandos.

Por exemplo, para somar 4 e 7, há duas possibilidades: você tanto pode usar a primitiva soma que exige dois argumentos: soma 4 7, ou você pode usar o operador +: 4+7.

Ambos têm o mesmo efeito.

A tabela abaixo mostra a equivalência entre operadores e primitivas:

soma +	diferença -	produto *	quociente /	ou 	e &	são iguais? =
soma 4 7 4 + 7	diferença 4 7 4 + 7	produto 4 7 4 + 7	quociente 4 7 4 / 7	3=4 ou (7 <= 49/7) 3=4 (7 <= 49/7)	3=4 e 7=49/7 3=4 & 7=49/7	3+4 = 7-1

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
```

Uso de Cores

As cores são definidas no XLogo com a ajuda de 3 números entre 0 e 255. Esse 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 esse sistema não é intuitivo o XLogo proporciona 16 cores pré-definidas 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]					

Eis três comandos que darão o mesmo efeito.

```
mudecf laranja
mudecf 13
mudecf [255 200 0]
```

Note que o nome da cor (laranja) ou seu número (13) não é precedido de aspa.

Primitivas

A tartaruga ("Tat") é controlada por comandos chamados "primitivas".

Elas são apresentadas nas seções a seguir:

Subseções

- | | |
|---|--|
| <ul style="list-style-type: none">• Movimento da Tat; configurações de fonte, lápis e cor• Tat no Espaço<ul style="list-style-type: none">• Um cubo e algo mais• Primitivas mostre e escreva• Aritmética e operações lógicas• Operações com listas• Booleanos• Testar uma Expressão• Lidar com procedimentos e variáveis• Acesso a arquivos• Função pinte e pintezone• Comandos de parada | <ul style="list-style-type: none">• Loops<ul style="list-style-type: none">• Um loop com repita• Um loop com para• Um loop com enquanto• Modo Tartarugas Múltiplas• Tocar música• Receber entrada (input) do usuário<ul style="list-style-type: none">• Interação com teclado e exemplos• Interação com mouse e exemplos• Tempo e data• Uso de Rede<ul style="list-style-type: none">• Como usar• As primitivas de Rede |
|---|--|

Movimento da Tat e Configurações

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

Primitivas	Argumentos	Uso
paradireita, pd	n: ângulo	Gira a Tat n graus para a direita em relação à direção que ela aponta.
paraesquerda, pe	n: ângulo	Gira a Tat n graus para a esquerda em relação à direção que ela aponta.
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	p i f	arco a b c desenha um arco de círculo de p passos ao redor da Tat entre os ângulos i (inicial) e f (final) dela (a Tat é o centro do círculo). Exemplo: arco 200 20 80
carregueimagem, carimg	a: lista	Carrega uma imagem na tela do XLogo. A posição da Tat será o canto superior esquerda 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 a Tat apontado 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 a 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 (V. mudedireção). Por exemplo: rotule [Que beleza!] escreverá a sentença "Que beleza!" onde quer 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 usando 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 desenhar 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 o 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 usado 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	Informa verd (o eixo está em exibição) ou falso
eiox?	nenhum	
eioy?	nenhum	
escondetat, dt	nenhum	Torna a Tat 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, use semgrade. Exemplo: grade 10 10
grade?	nenhum	Devolve verd (a grade está em exibição) ou falso (a grade não está em exibição).
invertelápis, il	nenhum	A Tat usa lápis no modo inverso, isto é, risca onde não tiver nada e apaga se já tiver.
lápisinta, lp	nenhum	A Tat usa lápis a riscar com sua cor clássica (preta).
limpedesenho, ld	nenhum	Limpa todos os desenhos na tela e restaura a Tat (coloca-a no centro).
lt, limpetexto	nenhum	Limpa (apaga) tudo que estiver escrito na linha de comandos 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"
mostretat, at	nenhum	Torna a Tat visível na tela.
mudecordoeixo	a: número inteiro ou lista [verm verde azul]	Muda a cor do eixo segundo a convenção de cores abaixo (mudecordofundo): mudecordoeixo [125 64 23] eixo 30
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.
mudecordagrade, mudecdg		Muda a cor da grade (veja acima, em mudecordofundo, para a convenção de cores). Exemplo: grade 30 30 mudecordagrade [125 64 23]
mudecordolápis, mudecl		Muda a cor do lápis. (mesma convenção de mudecordofundo)

mudeespessuradolápis, mudeel	n: número	Define a espessura (grossura) do lápis em pixels. 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 usar a primitiva rotule para escrever na área de desenho, é possível modificar o tamanho da fonte com mudefonte. O tamanho padrão da fonte 12.
mudenomefonte, 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).
mudequalidadedaimagem, mudeqi	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, use a primitiva muderoupa com números entre 0 e 6 (0 é um triângulo).
mudeseparação	a: número	Determine 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.
mudetamanhodajanela, mudetamjan	lista	Altera as dimensões da área de desenho. Exemplo <code>mudetamanhodajanela [650 850]</code> .
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 essa primitiva), 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 câmara que aparece à esquerda da janela de comandos já escritos.
pontadolápis, pl	nenhum	Informa a forma da ponta do lápis. 0 (quadrada) e 1 (redonda).
pos	nenhum	Informa a posição da Tat. Por exemplo: <code>pos</code> devolve [10 -100]
qualidadedaimagem, 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, eixo ou eixoy. Exemplo: <code>eixo 30</code>
semgrade	nenhum	Elimina a grade da área de desenho. Experimente <code>grade 10 10</code>
semlimite	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.
tamanhodajanela, 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	nenhum	Devolve o fator de zoom usado na área de desenho
mudezoom	a	Amplia (ou reduz) a área de desenho de acordo com o fator a escolhido. Exemplo: <code>mudezoom 2.5</code>

Tat no Espaço

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

A Técnica da Perspectiva

Para representar o espaço em três dimensões quando temos apenas duas (um plano), fazemos uso da perspectiva de projeção.

Uma câmara observa a cena 3D e a visão é projetada sobre um plano intermediário.

Ao lado, um esquema dessa situação:

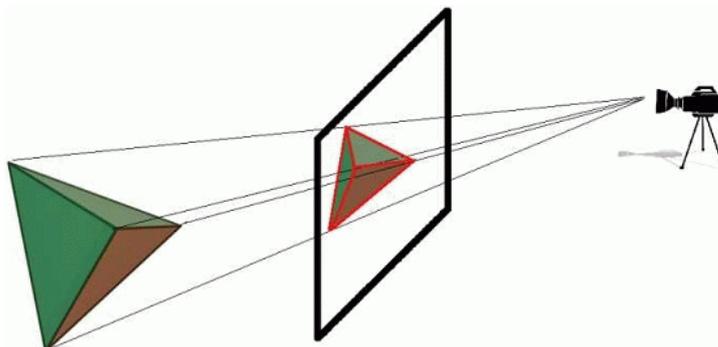


Figura 3: Técnica da Perspectiva

Certas primitivas nos permitem posicionar a câmara ao nosso modo, a tela de projeção situada numa distância igual à metade da distância câmara-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 3 valores de ângulos:

(1) **Rolagem** (Roulis ou Roll): Giro longitudinal da Tat (**eixo 0y**);

(2) **Arfada** (Tangage ou Pitch): Giro da Tat sobre sua largura (**eixo 0x**);

(3) **Guinada** (Cap ou Yaw): Giro da Tat em torno do **eixo 0z**.

Para tentar entender melhor, fazemos uma comparação com os giros ou manobras em um avião nas figuras abaixo (os nomes entre parentêses acima são os seus nomes em francês e inglês, respectivamente):

Figura 4: Analogias com avião:



Podem parecer complexos numa primeira vista mas você poderá notar muitas coisas continuam as mesmas [ou parecidas] do mundo plano. 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 4.a)
- roleparaesquerda (rolepe): a Tat gira n graus para a esquerda em seu eixo longitudinal (figura 4.a)
- levanteariz: a Tat ergue seu nariz n graus, ou seja, em seu eixo transversal (figura 4.b)
- baixeariz: a Tat gira seu nariz n graus para baixo, ou seja, em seu eixo transversal (figura 4.b)

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

```
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:

<pre>ld repita 4 [pf 100 pd 90] baixeariz 90 repita 4 [pf 100 pd 90]</pre>	<p>Figura 5: Quadrados em Perspectiva</p>
--	--

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

Entenda que as 3 primitivas de rotação são interligadas. Por exemplo, tente o seguinte:

```
ld
roleparaesquerda 90 levanteariz 90 roleparadireita 90
```

Primitivas para 2D e 3D

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

arco	centro	círculo	direçãopara	direção
distância	mudepos	mudedireção	mudex	mudey
ponto	pos, posição	rotule	tamanhorotule, tr	

Primitivas exclusivas para o modo 3D

- **mudexyz**: essa primitiva desloca a Tat até o ponto escolhido. Essa primitiva exige 3 argumentos que representam as coordenadas do ponto. Essa primitiva é similar a **mudepos** exceto que as coordenadas não são fornecidas em uma lista.
Exemplo: **mudexyz -100 200 50** moverá a Tat para o ponto $x = -100$; $y = 200$; $z = 50$.
- **mudez**: essa primitiva 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 3 números (os ângulos de rolagem, levanta/baixa nariz e de guinada). 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]. Note essa ordem de números. se por exemplo, a lista de valores for [100 20 90], significará que se você desejar a mesma orientação a partir da posição original (após um **limpedesenho**), você deverá escrever a seguinte sequência: **roleparadireita 100 levantarariz 20 paradireita 90**.
Se você inverter a ordem dessa instrução, você 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 abaixado).

Visor 3D

O xLogo possui um visor 3D desde a versão 0.9.92 que permite visualizarmos desenhos 3D. Esse módulo usa bibliotecas JAVA3D, de modo que é necessário ter o java3D 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 aquelas formas que desejamos registrar para uma futura visualização.. é possível guardar polígonos (superfícies), linhas, pontos ou textos.

Para usar esse recurso, temos as seguintes primitivas:

- **iniciepolígono**: fará com que os próximos movimentos da Tat sejam usados para criar um polígono.
- **terminepolígono**: encerra o polígono iniciado por **iniciepolígono**. O polígono desenhado poderá ser exibido pelo **visor3d**.
- **inicielinha**: fará com que os próximos movimentos da Tat sejam usados para criar uma linha representável no **visor3d**.
- **terminelinha**: finaliza a linha iniciada por **inicielinha**. Essa linha é representável no visor3d.
- **inicieponto**: fará com que os próximos movimentos da Tat sejam representado como um ponto no **visor3d**.
- **termineponto**: finaliza o ponto iniciado por **inicieponto**.
- **inicietexto**: permite que seja representado no **visor3d** qualquer texto escrito na área de desenho com a primitiva **rotule**.
- **terminetexto**: finaliza o texto iniciado por **inicietexto** para ser representado no **visor3d**.
- **visor3d** ou **mostrepolígono**: exibe o visor3d. Serão exibidos todos os objetos gravados na memória do xLogo por meio das primitivas apresentadas acima.
- Você poderá controlar o movimento da câmara com as seguintes opções:
 - 1) a cena será rotacionada se mover o mouse com o botão esquerdo pressionado.
 - 2) a cena será translada se mover o mouse com o botão direito pressionado.
 - 3) use o botão de rolagem do mouse para controlar a ampliação da cena (zoom).

Um Cubo e Algo Mais

Podemos imaginar o cubo em que cada face é um quadrado com 400 passos de tartaruga de lado. Eis uma solução:

```
aprenda quadrado
# o quadrado é desenhado e registrado como polígono
iniciepolígono
repita 4 [parafrente 400 paradireita 90]
terminepolígono
fim

aprenda cuboSimples
limpedesenho logo3d mudecordolápis amarelo
# desenhar as primeiras 4 faces
repita 4 [quadrado usenada paradireita 90 parafrente 400 paraesquerda 90
roleparadireita 90 uselápis]
# desenhar a face inferior
baixenariz 90 quadrado levantenariz 90
# desenhar a face superior
parafrente 400 baixenariz 90 quadrado
# visualizar
mostrepolígono
fim
```

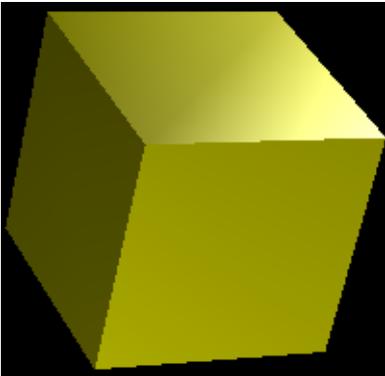


Figura 6: Um cubo

Digite o comando **cuboSimples**. Agora, é interagir com o cubo (lembre-se que cada botão do mouse ou rato tem a sua finalidade no visor 3d!).

O que ocorre se substituirmos as primitivas `iniciepolígono` e `terminepolígono` por, respectivamente, `inicielinha` e `terminelinha`?

Finalmente, o que ocorre se substituirmos essas primitivas por, respectivamente, `inicieponto` e `termineponto`? Experimente essas alterações.

Um exemplo simples do que foi mostrado até agora nesse manual:

```
logo3d
iniciepolígono repita 5 [pf 45 pd 72] terminepolígono
un pf 45 ul
inicielinha pf 45 pd 45 pf 45 terminelinha
inicietexto rotule "ok terminetexto
mostrepolígono
```

Eis algo com efeito interessante:

```
ld mudecf preto dt #limpa a tela, muda a cor de fundo e esconde a Tat
inicielinha
repita 6 [repita 24 [ul mudectl jf 125 lista cv 2 * cv círculo 45 rolepd 2 un pf 2
ul]]
terminelinha
mostrepolígono
```

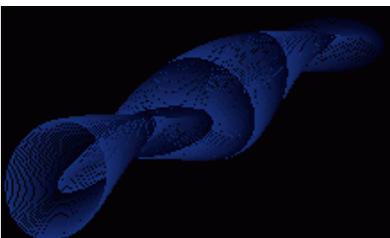


Figura 7: Uma figura interessante

* Luz

Você pode especificar 4 luzes no visor 3D. Por padrão, o cenário 3D possui apenas 2 pontos de luz habilitados ("duas lâmpadas"). Clique em um dos 4 botões com desenho de lâmpada disponíveis no visor 3D e uma caixa de diálogo aparecerá:

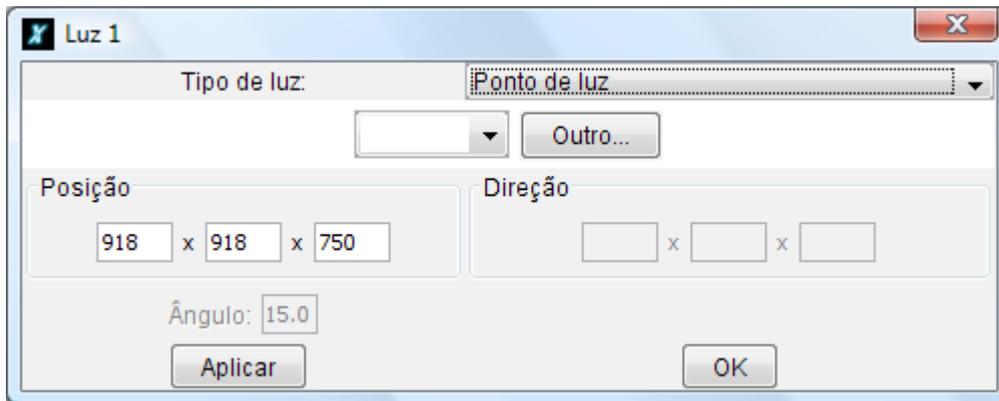


Figura 8a: Janela de diálogo para Luz

Vários tipos de luz estão disponíveis:

- Luz Ambiente:** luz uniforme, você necessita apenas especificar sua cor.
- Luz Unidirecional:** difusa de acordo com uma contante de direção. É o mesmo caso que *Ponto de Luz*, em que a fonte está muito distante do observador, como o Sol.
- Ponto de Luz:** possui uma posição específica. É semelhante ao farol dianteiro de um carro.
- Foco:** também é como o Ponto de Luz, exceto que se apresenta como um cone de luz. Você deve especificar um valor do ângulo para esse cone.

* Neblina

Você pode adicionar um efeito de neblina ao visor 3D. Clique no botão da nuvem que aparece no visor 3D que exibirá a seguinte caixa de diálogo:

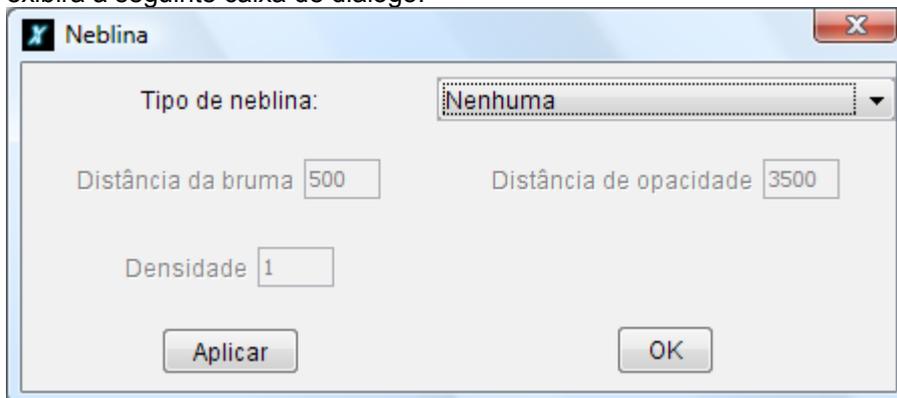


Figura 8b: Janela de diálogo para Neblina

Há dois tipos de neblina disponíveis:

- Neblina Progressiva:** a opacidade da neblina é progressiva. Necessita de dois parâmetros:
 - A distância em que a neblina inicia;
 - A distância em que a neblina termina;
- Neblina Densa:** Uma neblina uniforme para toda a cena. Você necessita apenas especificar sua densidade

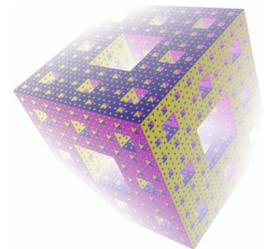


Figura 9: Exemplo de neblina progressiva

Escrever

Escrever na área de texto com as primitivas mostre ou escreva

Na tabela abaixo estão organizadas as primitivas que permitem ajustar as propriedades de texto. Tais primitivas controlam a cor e o tamanho na área de histórico, são 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: <i>esc "almo escreva "fada.</i> Compare com <i>mo "almo mo "fada</i>
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: <i>esc "almo escreva "fada.</i> Compare com <i>mo "almo mo "fada</i>
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: <i>nenhum, negrito, itálico, riscado, subescrito, sobreescrito, ou sublinhado.</i>
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 2 elementos. O primeiro é o número que corresponde à fonte usada 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] mudeestilo "itálico escreva "\ x mudeestilo "sobreescrito escreva 2 meu texto x².

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 ou 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 ou 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 ou verd ou falso.	mostre não 2 > 3

Eis uma lista de comandos relacionados a números:

Primitiva	Parâmetro	Uso	Exemplo
absoluto	abs	Devolve o valor absoluto de a	abs -23
aleatório	aleatório n	Sorteia um número inteiro positivo menor ou igual a n . V. primitiva <i>sorteie</i>	aleatório 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 . V. primitiva <i>aleatório</i>	sorteie 23
mudeprecisão	número inteiro positivo	Define o número de decimais a operar nos cálculos. Por padrão usa 16 decimais. Se usar um número negativo, xLogo adotará o padrão. Se por engano você fornecer uma decimal, o xLogo arredondará.	mudeprecisão 8
precisão	nenhum	Devolve o número de decimais em uso nos cálculos. O valor padrão é -1 (corresponde ao limite de 16 algarismos após a vírgula decimal).	mostre precisão
acos, arccoseno	acos a	Devolve o ângulo cujo coseno é a	acos 0.8
asen, arcseno	asen a	Devolve o ângulo (em graus) cujo seno é a	asen 0.5
atan, arctangente	atan a	Devolve o ângulo cuja tangente é a	atan 0.8
cos, coseno	cos a	Devolve o coseno de a	cos 45
sen, seno	sen a	Devolve o seno de a ("a" em graus)	sen 45
tan, tangente	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` (ie, 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, use a primitiva "menos" para especificar o número negativo - `mudexy 200 menos 10`.

Operações com Listas

Primitiva	Parâmetro	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	mostre conte [2 3 oi] mo conte "vila"
elemento, elem	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 for palavra, devolverá uma de suas letras. Sendo lista, devolverá 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]
junténofim, jf	a b	Coloca "a" no fim da lista b	mo jf 3 [7 oi]
junténoinício, 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 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"

Booleanos

Um booleano é uma primitiva que devolve a palavra "verd" ou a palavra "falso". Essas primitivas terminam com ponto de interrogação.

Primitiva	Parâmetro	Uso
éant? éanterior?	a b	Retorna <i>verd</i> se "a" for número, letra, ou palavra anterior a "b" (em ordem numérica ou alfabética). Exemplo: <code>mo éanterior? 23 42</code>
éinteiro?	a	Retorna <i>verd</i> se a for um número inteiro; <i>falso</i> , se não
élista? élista?	a	Retorna <i>verd</i> se a for uma lista
émembro? emembro?	a b	Retorna <i>verd</i> se "a" pertencer à lista ou palavra "b".
enum? énúmero?	a	Se a for um número, retornará <i>verd</i> . Do contrário, retornará <i>falso</i> .
eprim? éprimitiva?	a	Informa <i>verd</i> se "a" for uma primitiva do xLogo. <code>mo eprim? "parafrente</code>
eproc? éprocedimento?	a	Informa <i>verd</i> se "a" for um procedimento. Exemplo: <code>mo eproc? "quadrado</code>
éuselápis?	nenhum	Informa <i>verd</i> se o lápis estiver sendo usado (a Tat desenha ao se mover?)
évariável? evar?	a	Retorna <i>verd</i> se a for uma variável definida
evazio? évazio?	a	Retorna <i>verd</i> 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 <i>falso</i> . <code>mo membro "a "cidadão</code> <code>mo membro 3 [1 2 3 oi 5]</code>
epal? épalavra?	a	Se a for uma palavra, retornará <i>verd</i> . Do contrário, retornará <i>falso</i>
sãoiguais?	a b	Retorna <i>verd</i> se a e b forem iguais. Eg.: <code>mo sãoiguais? "c "C</code> <code>mo sãoiguais? "c "c</code>
verd	nenhum	Retorna <i>verd</i>
visível?	nenhum	Informa se Tat aparece (se está visível)

Testar uma Expressão

Como em toda a linguagem de programação, Logo lhe permite verificar se uma condição é satisfeita e então executar 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. De outro modo, 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]`

Lidar com Procedimentos e Variáveis

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

```
aprenda nome_do_procedimento :v1 :v2 :v3 ....  
  Corpo do procedimento  
fim
```

`nome_do_procedimento` é o nome dado ao procedimento.

`:v1 :v2 :v3` aguardam por variáveis usadas internamente por esse procedimento (variáveis locais).

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

Por exemplo:

```
aprenda quadrado :s  
  repita 4[pf :s pd 90]  
fim
```

O procedimento é chamado `quadrado` e leva um parâmetro chamado `s`.

`quadrado 100` desenhará um quadrado de comprimento 100 ([veja mais exemplos de procedimentos no final desse manual](#)).

Desde a versão 0.7c, é possível inserir comentários no corpo dos procedimentos desde que precedidos pelo símbolo #. Outros programas LOGO, tais como Superlogo, Megalogo e Micromundos usam o símbolo ; (ponto e vírgula). Nada do que é escrito à direita de # 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 de aprenda nem de fim.

Conceito de variáveis - Há dois tipos de variáveis:

- Variáveis Globais: aquelas que são acessíveis em qualquer parte do seu projeto (programa).
- Variáveis Locais: acessíveis somente no procedimento em que foram definidos.

Mais Primitivas

Nessa versão de LOGO, variáveis locais não são acessíveis em subprocedimentos. Ao final de um procedimento, as variáveis locais são apagadas.

Primitivas	Parâmetros	Uso
adeus	nenhum	Fecha o programa xLogo
atr, atribua	a b	Atribui o valor b para a variável a. Exemplo: atr "var 3 mo :var
atrlocal	a b	Cria a variável local "a" e lhe atribui o valor "b". atrlocal "a 2 atrlocal "r [oi, povo]
comandoexterno	lista	Executa um comando de sistema. A lista deve conter sublistsas que façam sentido para o seu sistema operacional. No Linux: comandoexterno [[gedit] [/home/xlogo/file.txt]] No Windows: comandoexterno [[notepad] [c:/file.txt]]
objeto	a: palavra	Mostra o valor da variável a.
def defina	pal1 lista1	Define o procedimento pal1, que requer a(s) variável(is) da lista2. As instruções do procedimento estão contido na lista3. def "poligono [[lados tam] [repita :lados [pf :tam pd 360 / :lados]]] O exemplo acima cria o procedimento poligono que desenhara um poligono regular quando escrever na linha de comandos algo como poligono 6 45 Note que lista1 dever ser constituída por pelo menos 2 sub-listas. A primeira define as variáveis. As demais representam cada linha do procedimento. defina "tarefa [[a b][escreva :a][escreva :b]] tarefa "música "dança músicadança mostre texto "tarefa [a b][escreva :a][escreva :b] Dúvida? Faça os exemplos acima e confira como aparece no Editor do xLogo.
texto	pal1	Lista o procedimento pal1 na forma esperada por Defina (confira esse procedimento logo acima) defina "tarefa [[a b][mostre :a][mostre :b] tarefa "música "dança música dança mostre texto "tarefa [[a b][mostre :a][mostre :b]]
el, elimine	a: palavra	Deleta (apaga) o procedimento de nome "a": el "poligono
et, eliminetudo	nenhum	Deleta (apaga) todas as variáveis e procedimentos atuais.
execute	a	Executa a lista de instruções contidas em a.
imts	nenhum	Lista todos os procedimentos atuais.
listear	nenhum	Devolve uma lista contendo as variáveis que você definiu
local	a b	Usada em procedimentos, cria a variável local "a". Para que ela tenha um valor, usa-se a primitiva atribua local "a atr "a 2 Pode-se também criar várias de uma só vez colocando-as entre colchetes: local [a b r] atr "a 2 atr "r [oi, povo] Consire também usar a primitiva atrlocal (ver no início dessa tabela).
mate	a	Deleta (apaga) a variável ^a
conteúdo	nenhum	Retorna uma lista formados de três outras. A primeira contém todos os procedimentos definidos; a segunda, todas as variáveis e a última todas as listas de propriedades.

É possível acompanhar o desenrolar de um procedimento. Este modo também permite ver que os comandos retornam com ajuda da primitiva saída. Para ativar este modo, basta escrever:

rastreie

Naturalmente, **parerastreie** desativará o modo «rastreie».

Ao lado, temos 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
```

Acesso a 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. Esse 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 a tartaruga está. Os formatos suportados são o png e o jpg. O caminho especificado deve ser parente ao diretório atual. Ex: carimg "tortue.jpg
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 efectiva apenas quando fecha-se o fluxo com a primitiva fechefluxo.
edite	lista ou palavra	Abre o Editor com todos os procedimentos especificados na lista (ou por palavra).
edtudo	nenhum	Abre o Editor com todos os procedimentos existentes.
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	guardado "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. Esse comando não funcionará com caminho absoluto.
guarde	w: palavra l:lista	Um bom exemplo para explicar isso: 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. Esse comando não funcionará com caminho absoluto.
guardeimagem	"arquivo [xmin ymin xmax ymax]	Guarda em arquivo a imagem da área de desenho. - Se o arquivo terminar em ".png", a imagem será gravada no formato png - Se o arquivo terminar em ".jpg" ou ".jpeg", a imagem será gravada no formato jpeg - Se não escrever nada, a imagem será gravada no formato png e ".png" será adicionado no final do nome do arquivo. "xmin xmax ymin ymax" permite escolher os dois cantos opostos do retângulo que corresponde à área de seleção. Se a lista estiver vazia ou conter erros, será usada toda a área da imagem.
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 caracter neste arquivo. Esta primitiva retorna um número que representa o valor do carácter (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, mudediretório	l: lista	Especifica o diretório a usar. O caminho deve ser absoluto. O diretório tem que ser especificado dentro de uma lista e o caminho não pode conter espaços.
comandoexterno	lista1	Permite executar um comando de sistema através do xLogo. lista1 deve conter várias listas contendo cada uma as palavras que constituem o comando. Alguns exemplos: comandoexterno [[gedit] [/home/xlogo/arquivo.txt]] lança o aplicativo gedit (editor de texto comum ao Gnome do Linux) e carrega o arquivo /home/xlogo/arquivo.txt comandoexterno [[notepad] [C: /arquivo.txt]] lança o aplicativo notepad ("bloco de notas" do Windows) e carrega o arquivo nomeado C: /arquivo.txt Esta sintaxe ligeiramente especifica permite nomeadamente a utilização de espaços nos caminhos de arquivos.

Segue um exemplo de utilização das primitivas que permitem ler e escrever em arquivos. Apresentaremos este exemplo para uma arquitetura de tipo Windows. Os outros utilizadores saberão adaptá-lo, pois a ideia é a mesma. O objetivo é criar o arquivo `c:\exemplo.txt` contendo as três linhas:

```
ABCDEFGHIJKLMNOPQRSTUVWXYZ
abcdefghijklmnopqrstuvwxyz
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 [ABCDEFGHIJKLMNOPQRSTUVWXYZ]
```

```
escrevalinha 2 [abcdefghijklmnopqrstuvwxyz]
```

```
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]
```

Lê-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]`

Pinte e PinteZona

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

Vejamos na forma de exemplo para explicar a diferenças entre aquelas duas primitivas: a Tat está sobre um ponto de cor branca.

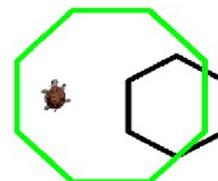


Figura 10: Um desenho a ser pintado

A primitiva `pinte` colorirá todos os pixels brancos com a cor do lápis que a Tat está a usar.

Se, por exemplo, escrevermos: `mudecl 1 pinte`

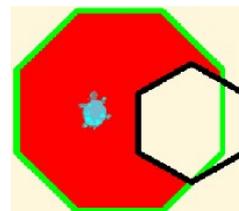


Figura 11: Uso da primitiva `pinte`

Voltemos ao primeiro caso (a figura 10): se a cor do lápis da Tat for preta, a primitiva `pinteZona` colorirá toda a região até encontrar cores diferentes daquela ela estava sobre (neste exemplo, ao atingir as linhas preta ou verde).

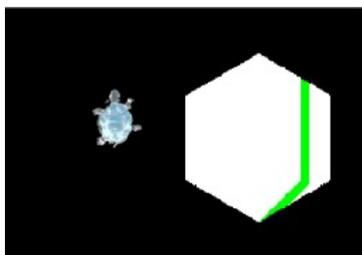


Figura 12: Uso da primitiva pintezona

A primitiva pinte permite preencher uma área delimitada. Ela pode ser comparada com o "baldinho" disponível em muitos editores de imagem (paint, tuxpaint, gimp, etc.).

Há duas regras a considerar ao usar essa primitiva:

1. O lápis deve estar em uso (ul).
2. A Tat não pode estar sobre um pixel com a mesma cor que será usada para preencher a área (se desejar pintar algo de vermelho, a Tat não pode estar sobre um ponto vermelho...)

Eis um exemplo:

```
aprenda rainbow :c
```

```
se :c<100 [pare]
```

```
semicirc :c pd 180 pf 20 pe 90
```

```
rainbow :c-40
```

```
fim
```

```
aprenda semicirc :c
```

```
# desenha um semi-círculo com
```

```
diâmetro :c
```

```
repita 180 [pf :c*tan 0.5 pd 1]
```

```
pf :c*tan 0.5
```

```
pd 90 pf :c
```

```
fim
```

```
aprenda dep
```

```
un pd 90 pf 20 pe 90 ul
```

```
fim
```

```
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
```

```
vai
```

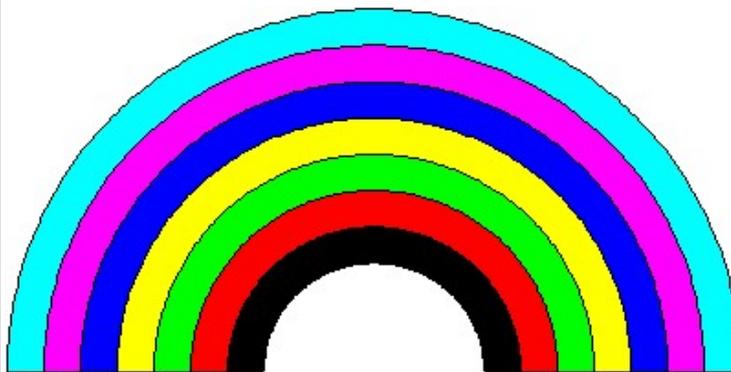


Figura 13: Arco-em-LOGO

PreenchaPolígono

Essa primitiva preenche (pinta) uma série de triângulos. Assim, cada vez que desenha o segundo segmento de reta, o xLogo pintará o triângulo formado pelos 3 vértices. Uma lista conterà as instruções a desenhar a forma a ser preenchida.

Vejamos na forma de exemplos.

1. Para preencher um quadrado:

```
ld PreenchaPolígono [repita 4 [parafrente 100 paradireita 90]]
```

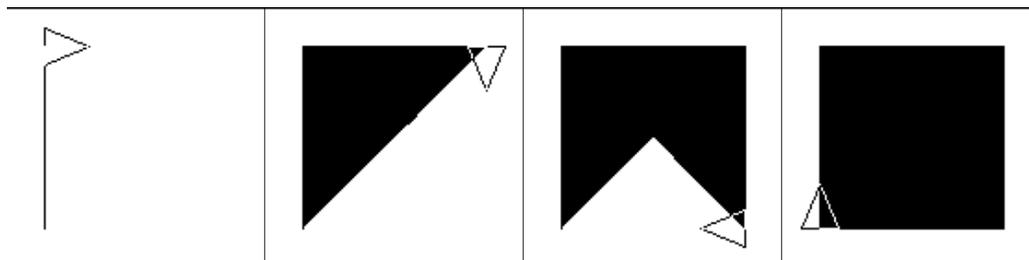


Figura 12a: Pintar um quadrado (passo a passo)

2. Para preencher uma estrela de 5 pontas:

```
repita 5 [parafrente 100 PreenchaPolígono [paratrás 100 paradireita 144 parafrente 100 ] paratrás 100 paraesquerda 72]
```

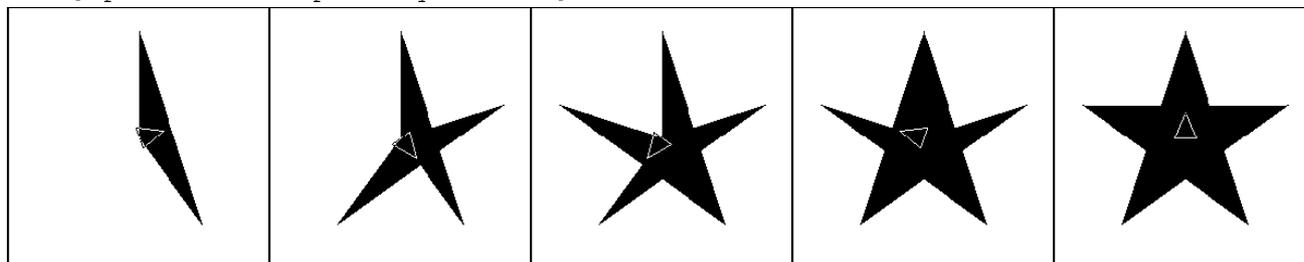


Figura 12b: Pintar um estrela (passo a passo)

Comandos de Parada

xLogo possui três comandos de parada: pare, paretudo e saída.

- **pare** pode ter dois resultados. Se incluso em um loop repita ou enquanto, o programa sai do loop. Se isso ocorrer em um procedimento, o programa interrompe o procedimento de imediato.
- **paretudo** interrompe definitivamente a execução de todos os procedimentos em curso.
- **saída** (ou **devolva**) permite parar um procedimento com um valor a ser devolvido.

Veja várias situações de uso dessas primitivas nos exemplos no final dessa documentação.

Loops

xLogo possui 7 primitivas que permitem construir loops (laços): *repita*, *para*, *enquanto*, *paracada*, *parasempre*, *repitaenquanto* e *repitaaté*.

"Loop" é um termo usado em programação para indicar um procedimento ou conjunto de instruções em um programa de computador que são executadas repetidamente até que uma condição específica seja satisfeita ou até que o programa seja concluído.

Subseções

- [Um loop com repita](#)
- [Um loop com para](#)
- [Um loop com enquanto](#)
- [Um loop com paracada](#)
- [Um loop com parasempre](#)
- [Um loop com repitaaté](#)
- [Um loop com repitaenquanto](#)

Repita

```
repita n listadecomandos
```

n é um número inteiro e listadecomandos é uma lista contendo os comandos a executar. O interpretador LOGO irá implementar os comandos na lista n vezes: isso evita ter que escrever n vezes os mesmos comandos!

Por exemplo:

```
repita 4 [parafrente 100 paraesquerda 90] # Um quadrado de lado 100
repita 6 [parafrente 100 paraesquerda 60] # Um hexágono de lado 100
repita 360 [parafrente 2 paraesquerda 1] # Um, digamos, "360-gono" de lado 2
# De certa forma, quase um círculo!
```

Em laços com a primitiva repita podemos usar a variável interna contevezes. Ela designa o número de iterações em andamento (começando pelo número 1).

Por exemplo:

```
repita 3 [escreva contevezes]
1
2
3
```

Para

A sintaxe de para:

```
para [lista1] [lista2]
```

A primitiva "para" equivale ao comando "for" de outras linguagens de programação. Ela consiste em atribuir a uma variável diversos valores compreendidos num intervalo dado de acordo com um incremento definido.

A lista 1 contém três parâmetros: o nome da variável, o limite inicial, o limite final. Pode-se acrescentar um quarto argumento opcional que designa o incremento (de quanto em quanto a variável será incrementada). Se for omitido, o valor do incremento será 1.

Um exemplo:

```
para [i 1 4] [mo :i*2]
2
4
6
8
```

```
# No exemplo abaixo faz-se variar i entre 7 e 2
decrecendo de 1.5 cada ve
# note que o incremento usado é negativo
# Na lista 2, pede para exibir uma lista com o valor de i e
seu valor elevado à potência 2.
```

```
para [i 7 2 -1.5] [mo lista :i potência :i 2]
7 49
5.5 30.25
4 16
2.5 6.25
```

Enquanto

Eis a sintaxe para enquanto:

```
enquanto [listaaseravaliada] [listadecomandos]
```

listaaseravaliada é uma lista contendo um conjunto de instruções o qual pode ser avaliado como booleano (verdadeiro ou falso).

listadecomandos é uma lista contendo os comandos a serem executados. O interpretador LOGO continuará implementando listadecomandos enquanto a *listaaseravaliada* retornar verd (verdadeiro).

Por exemplo:

```
enquanto ["verd] [pd 1] # A Tat ficará dando voltas  
  
# Um exemplo que permite soletrar o alfabeto de trás para diante  
atr "lista "abcdefghijklmnopqrstuvwxy  
enquanto [não évazio? :lista] [mo último :lista atr "lista semúltimo :lista]
```

Parasempre

Eis a sintaxe de parasempre:

```
parasempre [listadecomandos]
```

Repete indefinidamente uma lista de instruções (executa para sempre o que for pedido).

Exemplo:

```
parasempre [pf 1 pd 1] # A Tat dará voltas indefinidamente [sem ficar tonta, rs, rs].
```

Para interromper a execução, clique no botão Pare do xLogo (aquele que fica na parte inferior direita da janela).

Paracada

Eis a sintaxe para paracada:

```
paracada nomedavariavel listaoupalavra listadecomandos
```

A variável assume um valor de cada item de uma lista, ou o caracter de uma palavra.

Os comandos (instruções) serão repetidos para cada valor que a variável assumir.

Exemplos:

```
paracada "i [a b c] [mostre :i]
```

a
b
c

```
paracada "i ["XLOGO] [mostre :i]
```

X
L
O
G
O

Repitaaté FaçaAté

Eis a sintaxe para essa primitiva:

```
RepitaAté [listadecomandos] [condição]
```

Repete instruções até a condição ser verdadeira. Os comandos (instruções) serão repetidos enquanto a condição for falsa.

Confira o exemplo ao lado.

```
atribua "i 0
```

```
RepitaAté [mostre :i atribua "i :i+1] [:i>4]
```

0
1
2
3
4

RepitaEnquanto FaçaEnquanto

Eis a sintaxe para essa primitiva:

```
RepitaEnquanto [listadecomandos] [condição]
```

Repete instruções enquanto uma condição for verdadeira.

Os comandos (instruções) serão repetidos enquanto a condição for verdadeira. A principal diferença para a primitiva enquanto é que o bloco de instruções é pelo menos uma vez mesmo que a condição seja falsa.

Por exemplo:

```
atribua "i 0
```

```
RepitaEnquanto [mostre :i atribua "i :i+1] [:i<4]
```

0
1
2
3
4

Múltiplas Tartarugas

É possível ativar várias tartarugas na tela. Por padrão, ao iniciar o Xlogo há somente uma Tat. Seu número é 0. Se desejar "criar" outra Tat, você pode usar a primitiva `atat` seguido pelo número da tartaruga. Para evitar estorvos, a Tat é criada no centro e é invisível. (você pode usar `at` ou `mostretat` para exibi-la). Então, a nova Tat estará ativa e obedecerá todas as primitivas enquanto você não mudar a tartaruga ativa com `atat`.

Você define o número máximo de tartarugas no menu Ferramentas - Preferências - Guia opções ou com a primitiva `mudemáximodetartarugas`.

Eis as primitivas para o modo múltiplas tartarugas:

Primitivas	Argumentos	Uso
<code>atat</code> , <code>atençãotat</code>	a: número	A Tat número a é a que está ativa agora. Por padrão, a Tat ativa ao iniciar o Xlogo é a de número 0.
<code>máximodetartarugas</code> , <code>maxtat</code>	nenhum	Defina o número máximo de tartarugas na tela no modo multitartarugas.
<code>mudemáximodetartarugas</code> , <code>mudemaxtat</code>	a: número inteiro	Devolve o número máximo de tartarugas na tela no modo multitartarugas.
<code>tativa</code>	nenhum	Devolve o número da tartaruga ativa.
<code>tartarugas</code>	nenhum	Devolve uma lista com números de todas as tartarugas existentes.
<code>semtat</code> , <code>semtartaruga</code>	a: número	Elimina a Tat número a

Tocar Música

Primitivas	Argumentos	Uso
<code>seq</code> , <code>sequência</code>	a: lista	Coloca na memória a sequência da lista. Leia após essa tabela sobre como escrever uma sequência.
<code>delseq</code> , <code>deletesequência</code>	nenhum	A sequência é deletada (apagada) da memória.
<code>indseq</code> , <code>índicesequência</code>	nenhum	Devolve a posição do cursor na sequência em uso.
<code>instr</code> , <code>instrumento</code>	nenhum	Devolve o número que corresponde ao instrumento em uso.
<code>mudeinstr</code> , <code>mudeinstrumento</code>	a: número	O instrumento em uso passa a ser o de número a. Veja a lista completa de instrumentos no menu Ferramentas-Preferências-Guia Som.
<code>mudeindseq</code> , <code>mudeíndicesequência</code>	a: número	Põe o cursor no índice a na sequência memorizada.
<code>toque</code>	nenhum	Toca a(s) sequência(s) memorizadas.
<code>toquemp3</code>	arquivo	Toca um arquivo mp3 que estiver no diretório atual ou na internet: <code>toquemp3 "arquivo.mp3"</code> <code>toquemp3 "http://meusite.com.br/arquivo.mp3"</code>
<code>paremp3</code>	nenhum	Interrompe (para) de tocar o arquivo chamado por <code>toquemp3</code>

Se deseja tocar uma música, você deve colocar as notas na memória por meio de uma lista chamada "sequência". Para criar a sequência, você pode usar a primitiva `seq` ou `sequência`.

Eis as regras para criar uma sequência válida:

dó ré mi fá sol lá si: as notas da primeira oitava.

Para um "ré" mais agudo, escreve-se `ré +`. Para um "ré" mais grave, escreve-se `ré -`.

Para subir ou descer uma oitava, use o símbolo ":" seguido por + ou -. Por exemplo, após `++` na sequência, todas as notas serão tocadas duas oitavas acima (dois ++).

Por padrão, as notas são tocadas com duração um (1). Se desejar alterá-la, escreva o número que corresponde à duração das notas. Por exemplo, `seq [sol 0.5 lá si] toque`, tocará sol com a duração 1 e lá e si com duração 0.5 (duas vezes mais rápido).

Que tal experimentar o seguinte exemplo?

```
aprenda tabac
# cria uma sequência de notas
seq [0.5 sol lá si sol 1 lá 0.5 lá si 1 :+ dó dó :- si si 0.5 sol lá si sol
    1 lá 0.5 lá si 1 :+ dó ré 2 :- sol ]
seq [:+ 1 ré 0.5 ré dó 1 :- si 0.5 lá si 1 :+ dó ré 2 :- lá ]
seq [:+ 1 ré 0.5 ré dó 1 :- si 0.5 lá si 1 :+ dó ré 2 :- lá ]
seq [0.5 sol lá si sol 1 lá 0.5 lá si 1 :+ dó dó :- si si 0.5 sol lá si sol
    1 lá 0.5 lá si 1 :+ dó ré 2 :- sol ]

fim
```

Para escutar a música, escreva na caixa de comandos: `tabac toque`

Agora, veremos uma aplicação interessante da primitiva `mudeindseq`. Escreva os comandos:

```
delseq          # Deleta a sequência memorizada
tabac           # Coloca as notas na memória
mudeindseq 2    # Pula as duas primeira notas (depois do "lá", nesse exemplo).
tabac           # Coloca na memória a mesma sequência porém puladas as 2 primeiras notas.
toque          # Toque a música para ver o efeito!
```

Você pode escolher o instrumento com a primitiva `mudeinstr` ou no menu Ferramentas-Preferências-Guia Som. Nessa guia você encontra a lista completa dos instrumentos disponíveis e seus respectivos números.



Figura 14: Uma partitura

Receber Entrada de Usuário

Subseções

- [Interação com o teclado e alguns exemplos de aplicação;](#)
- [Interação com o mouse e alguns exemplos de aplicação](#)
- [Interação por IG](#)

Interação com Teclado

No momento, textos do usuário são recebidos por meio de 3 primitivas: `étecla?`, `leiacar` e `leia`.
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:

Tabela 1: Valores de teclas

A --> 62	B --> 63	C --> 64	etc ...	Z --> 90	--> 37 ou 226 (NumPad)
--> 38 ou 224	--> 39 ou 227	--> 40 ou 225	Esc --> 27	Shift --> 16	Ctrl --> 17
F1 --> 112	F2 --> 113	etc ...	F12 --> 123	Espaço --> 32	Enter --> 10

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. A resposta será armazenada na forma de uma lista na variável :palavra, e será avaliada ao clicar em OK.`

Exemplo

```
leia lista [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

Interação com Mouse

No momento, há duas primitivas que tratam eventos com mouse: `leiamouse`, `mouse?` e `posmouse`.

`leiamouse`: o procedimento é interrompido até o usuário movimentar 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 ... (os botões são numerados da esquerda para a direita, em princípio).

`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.

Nesse 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
```

Nesse segundo exemplo, mas 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, desenha-se um quadrado de lado 40. Ao clicar com o botão esquerdo no botão direito, desenha-se um pequeno círculo. Por fim, ao clicar com o botão direito 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 [255 153 153]
pinte pt 10 pe 45 ul mudecl 0
fim

aprenda mouse
# o valor de leiamouse é armazenado na variável ev
atr "ev leiamouse
# 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>0 & :y<50 [
  se :ev=1 [circule]
  se :ev=3 [pare]]
mouse
fim
```

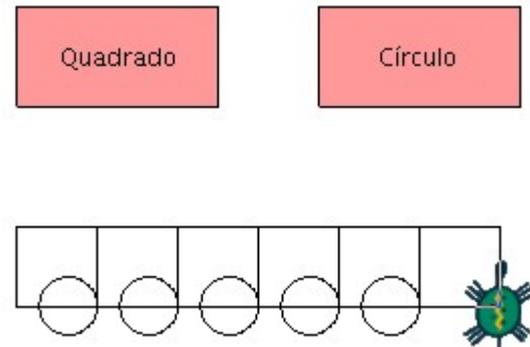


Figura 15: Uso do mouse

```
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
```

Interação por IG (Interface Gráfica)

XLogo lhe dá a possibilidade de criar alguns componentes gráficos na área de desenho (Botão, menu, ...). Esses componentes estão associados às interfaces gráficas e, para facilitar, todas essas primitivas iniciam por «IG»(Interface Gráfica). São as primitivas igbotão e igmenu e iglocal, e suas complementares igfaz, igapague e igdesenhe.

Para criar um componente: Para usar 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 conterá 3 entradas: item1, item2 e item3
igmenu "m [item1 item2 item3]
```

Como atribuir propriedades a esses 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, deleta) 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: Mostrará o componente gráfico na área de desenho. É importante, pois no xLogo o botão ou menu só será exibido com este comando. Por exemplo, para vermos o botão **b**:

```
igdesenhe "b
```

Como exemplo, vamos reescrever os procedimentos criados em [Interação com o mouse](#). Note que dessa vez o projeto fica bem mais econômico, por assim dizer.

```
aprenda vai                                aprenda botão
limpedesenho botão                          #cria a botão "bQ" com o rótulo "Quadrado"
un mudedos [0 -100] ul                       igbotão "bQ "Quadrado
fim                                           #Posiciona o botão "bQ"
                                              iglocal "bQ [-80 100]
aprenda circule                              #Informa ao botão bQ que ele executará o
repita 90 [pf 1 pe 4] pe 90 un pf 40 pd 90 ul procedimento quadrado ao ser clicado
fim                                           igfaz "bQ [quadrado]
                                              #Exibe o botão bQ
aprenda quadrado                             igdesenhe "bQ
repita 4 [pf 40 pd 90] pd 90 pf 40 pe 90    #A seguir, a mesma ideia geral para o botão "bC" que
fim                                           desenhará um círculo:
                                              igbotão "bC "Circulo
                                              iglocal "bC [80 100]
                                              igfaz "bC [circule]
                                              igdesenhe "bC
                                              fim
```

Tempo e Data

XLogo possui muitas primitivas para data, tempo ou contadores.

espere	n: número	Interrompe 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 3 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:

```
aprenda relógio
# Exibe o tempo no formato numérico
# Atualiza a cada 5 segundos
se fimcontador? [
  ld
  mudefonte 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 ]
relógio
fim
```

Uso de Rede

Uso de rede ("network") com xLogo: você pode inclusive iniciar um bate-papo (chat) em rede!

Subseções

- [Como usar](#)
- [As primitivas para Rede](#)

Como Usar

Antes de usarmos as primitivas de rede do xLogo, precisamos de uma pequena introdução aos princípios da comunicação em rede.

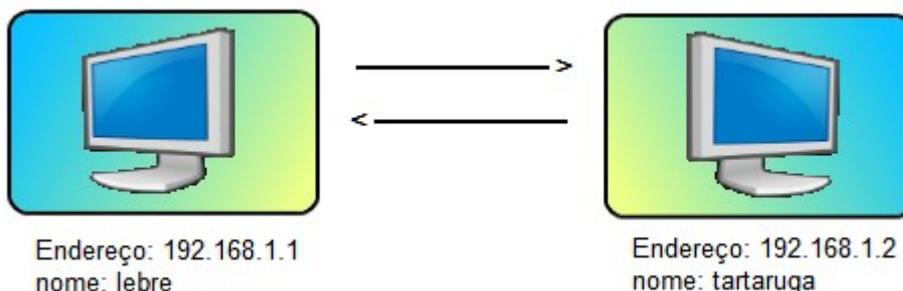


Figura 16: Uma rede simples

Dois ou mais computadores podem se comunicar em rede se ambos tiverem uma placa ethernet. Cada computador é identificado por um endereço particular chamado de endereço IP. Esse endereço IP consiste de 4 números inteiros, cada um entre 0 e 255 e separados por pontos. Por exemplo, O endereço IP do primeiro computador na ilustração acima é 192.168.1.1

Uma vez que não é fácil lembrar esses números, também é possível identificar cada número por um nome. Como podemos ver na figura, podemos nos comunicar com o computador da direita usando o endereço IP: 192.168.1.2, ou pelo seu nome: tartaruga

Por enquanto, acrescentarei só mais uma coisa. O computador local no qual você está trabalhando é localizado pelo endereço: 127.0.0.1. Nome geral é localhost. Veremos isso mais adiante na prática.

Primitivas de Rede

XLogo possui 4 primitivas que lhe permitem se comunicar via rede (network): `escutetcp`, `executetcp`, `chattcp` e `envietcp`. Nos próximos exemplos, usaremos o caso de dois computadores conforme foi mostrado na figura anterior.

- **escutetcp**: essa primitiva é a base de toda a comunicação por rede. Não necessita nenhum argumento. Ao executar essa primitiva em um computador, ele ficará a espera de instruções enviadas por outros computadores na rede (ficará na "escuta", por assim dizer).
- **executetcp**: essa primitiva permite a execução de instruções por um computador em rede. Sintaxe: `executetcp palavra lista`. A palavra pode ser o endereço IP ou nome do computador, a lista contém instruções a serem executadas.
Exemplo: Estou no computador lebre e quero desenhar um quadrado de lado 100 em outro computador. Assim, no computador tartaruga, deverei usar o comando `escutetcp`. No computador lebre, eu escrevo:

```
executetcp "192.168.1.2 [repita 4[pf 100 pd 90]]  
ou ainda  
executetcp "tartaruga [repita 4[pf 100 pd 90]]
```
- **chattcp**: Permite bate-papo ("chat") entre dos computadores em uma rede. Em cada computador mostrará uma janela de bate-papo. Sintaxe: `chattcp palavra lista`. A palavra pode ser o endereço IP ou nome do computador, a lista contém a sentença (frase) a exibir.
Exemplo: lebre quer falar com tartaruga.

Primeiro tartaruga executa `escutetcp` para que aguarde instruções de computadores em rede. Então lebre escreve: `chattcp "192.168.1.2 [Olá, tartaruga!]`.

As janelas de bate-papo serão abertas em ambos os computadores, permitindo que se converse por meio deles.

- **envietcp**: Envia dados por um computador em rede e retorna sua resposta. Sintaxe: `envietcp palavra lista`. A palavra pode ser o endereço IP ou nome do computador, a lista contém os dados a enviar. Quando o Xlogo for iniciado no outro computador, será respondido OK. É possível com essa primitiva se comunicar com um robô por meio de sua interface de rede. Então, a resposta do robô poderá ser diferente.

Exemplo: tartaruga quer enviar para lebre a sentença ("frase") "pi vale 3.14159".

Primeiro executamos em lebre a primitiva `escutetcp` de modo que fica a espera da comunicação de outro computador. Então, escrevemos em tartaruga: `mostre envietcp "lebre [pi vale 3.14159]`.

Uma pequena dica: inicie duas vezes o xLogo no mesmo computador.

- Na primeira janela, execute `escutetcp`.
- Na segunda, escreva `executetcp "127.0.0.1 [pf 100 pd 90]`

Assim, você moverá a tartaruga em outra janela! (rs, rs, isso é possível porque 127.0.0.1 designa seu endereço local, ou seja, é o seu próprio computador...).

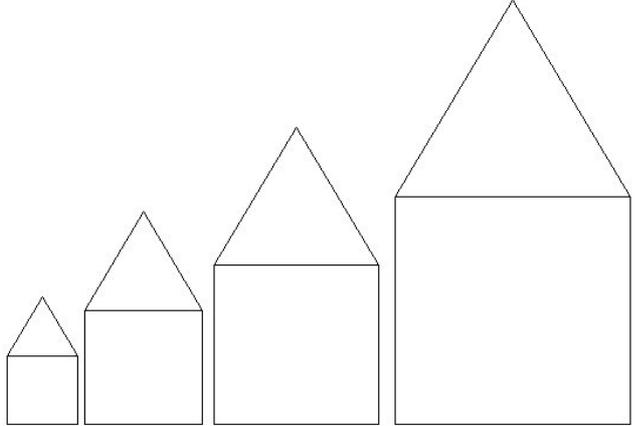
Exemplos de Programas

Subseções

- Desenhar casas
- Desenhar um retângulo preenchido
- Fatorial
- O floco de neve (agradecimento a Georges Noël)
- Um pouco de escrita...
- E conjugação...
 - Primeira versão
 - Segunda tentativa
 - Ou ainda: Uma pequena recursão !
- Tudo sobre cores
 - Vamos praticar!

- E se quiser um negativo??
- Um bom exemplo de uso de listas (agradecimento a Olivier SC)
- Uma linda rosácea

Desenhar Casas

 <p>Figura 17: Casas</p>	<pre> 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 </pre>
--	---

Desenhar um Retângulo Preenchido

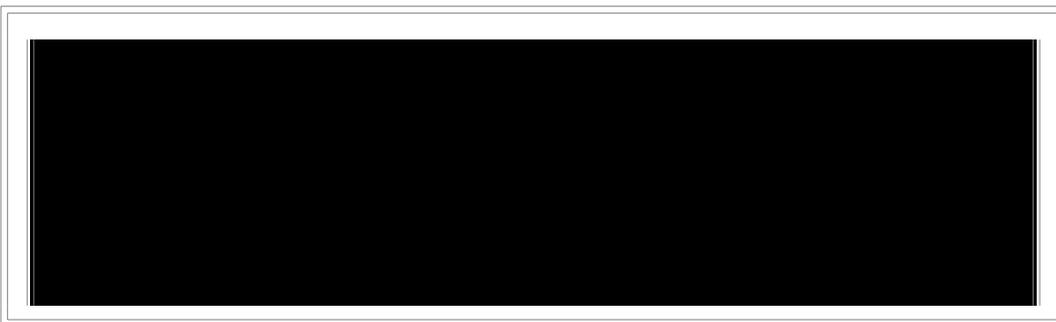


Figura 18: Retângulo

<pre> 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 </pre> <p>O símbolo na segunda linha equivale à primitiva <i>ou</i>.</p>	<pre> 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 </pre>
---	--

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

```

Fatorial

Lembre-se que:

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

<pre> aprenda fac :n se :n=1[saída 1][saída :n*fac :n-1] fim </pre>	<pre> mostre fac 5 120.0 mo fac 6 720.0 </pre>
---	--

Floco de Neve (agradecimento a Georges Noël)

```
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 floco Koch :ordem :len
repita 3 [pd 120 koch :ordem
:len]
fim
```

```
floco Koch 5 450
```

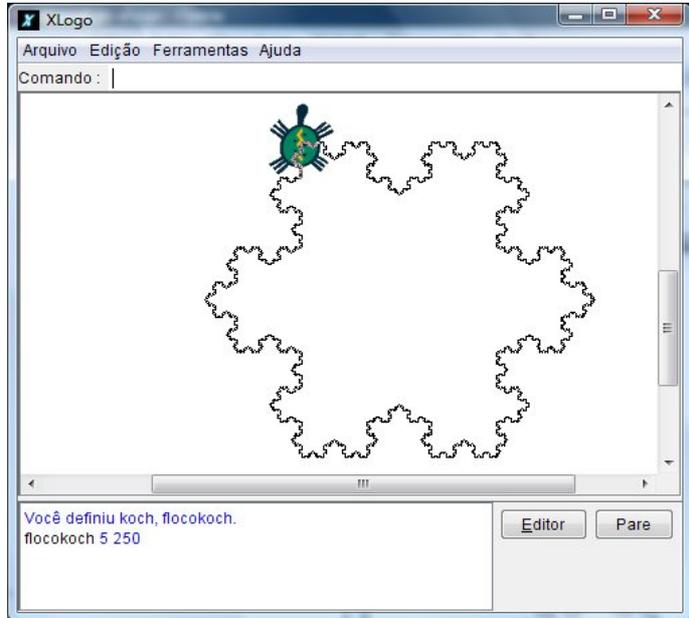


Figura 19: um fractal (o floco de neve)

Um Pouco de Escrita

aprenda escrever

```
dt repita 40[pf 30 pd 9 mudecl sorteie 7 rotule [XLogo é show de bola!]]  
fim
```

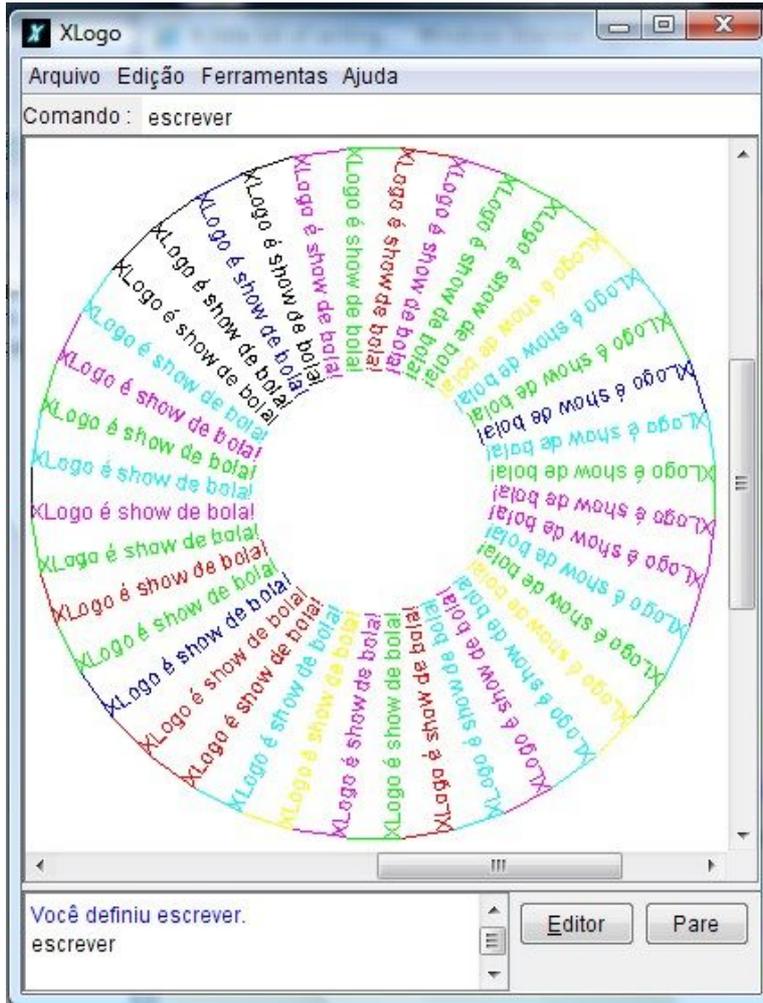


Figura 20: XLogo é show de bola!

E Conjugação...

Subseções

- [Primeira versão](#)
- [Segunda versão](#)
- [Ou ainda: Uma pequena recursão !](#)

Primeira Versão

Conjugar um verbo em francês

```
aprenda Fr_futuro :pal  
mo sn "je palavra :pal "ai  
mo sn "tu palavra :pal "as  
mo sn "il palavra :pal "a  
mo sn "nous palavra :pal "ons  
mo sn "vous palavra :pal "ez  
mo sn "elles palavra :pal "ont  
fim
```

```
Fr_futuro "parler  
je parlerai  
tu parleras  
il parlera  
nous parlerons  
vous parlerez  
elles parleront
```

Segunda Versão

Conjugação de verbos (exemplo em francês)

```
aprenda fut :pal
atr "pronomes [je tu il nous vous elles]
atr "terminações [ai as a ons ez ont]
atr "i 0
repita 6[atr "i :i+1 mo sn elem :i :pronomes pal :pal elem :i :terminações]
fim
```

fut "parler

je parlerai
tu parleras
il parlera
nous parlerons
vous parlerez
elles parleront

Conjugação de verbos (exemplo em português)

```
aprenda PTfut :pal
atr "pronomes [eu tu ele nós vós eles]
atr "terminações [ei ás á emos eis ão]
atr "i 0
repita 6[atr "i :i+1 mo sn elem :i :pronomes pal :pal elem :i :terminações]
fim
```

PTfut "falar

eu falarei
tu falarás
ele falará
nós falaremos
vós falareis
eles falarão

Uma Pequena Recursão!

```
aprenda fu :verbo
atr "pronomes [je tu il nous vous elles]
atr "terminação [ai as a ons ez ont]
conjuguar :verbo :pronomes :terminação
fim
```

```
aprenda conjuguar :verbo :pronomes
:terminação
se évazio? :pronomes [pare]
mo sn pri :pronomes pal :verbo pri
:terminação
conjuguar :verbo semprimeiro :pronomes
semprimeiro :terminação
fim
```

fu "parler

je parlerai
tu parleras
il parlera
nous parlerons
vous parlerez
elles parleront

Primeiro, algumas explicações: você notará que o comando `mudecl` aceita tanto uma lista quanto um número. Aqui estamos interessados em valores do código RGB (iniciais em inglês para vermelho, verde e azul). Cada cor no XLOGO é codificada usando valores (de 0 a 255) para vermelho, verde e azul. Cada uma dessas três cores recebe um valor de 0 a 255 para formar uma lista para a primitiva `mudecl`. As cores exibidas no monitor do computador são uma mistura dessas três cores em diferentes intensidades. Talvez não seja muito intuitivo, mas é basicamente assim que as coisas funcionam. Pense que todas as cores possíveis de serem exibidas pelo computador são formadas pela combinação do vermelho, verde e azul em diferentes intensidades. Valores quanto mais próximos de 0 dão tonalidades mais intensas. O azul puro, por exemplo, seria `mudecl [0 0 255]`. Para o preto, `mudecl [0 0 0]`. Para o branco, `mudecl [255 255 255]`.

Você também pode escolher cores pelo menu Ferramentas--> Mudar cor do lápis. Há três modos de fazê-lo nesse menu.

Por meio desse código, é fácil transformar uma imagem. Por exemplo, se desejar que uma foto colorida se transforme em uma imagem em preto e branco, você pode trocar cada pixel da imagem pela média aritmética dos valores desses 3 componentes [r g b]. Imagine que a cor de um ponto da imagem corresponda a [0 100 800]. Você pode calcular a média desses 3 números:

$$\frac{0+100+80}{3} = 60$$

, e a cor [60 60 60] deverá ser atribuída a esse pixel. Essa operação deverá ser repetida para cada ponto da imagem.

Subseções

- [Vamos praticar!](#)
- [E se desejar um negativo??](#)

Vamos Praticar!

Transformaremos em tons de cinza uma imagem de 100 x 100 pixels. Isso significa que teremos 100 x 100 = 10000 pixels a serem modificados.

Vamos à tarefa: primeiro, nos referiremos ao canto esquerdo superior da imagem como sendo [0 0]. A Tat examinará os primeiros 100 pixels da primeira linha, seguidos pelos primeiros 100 pixels da segunda linha, e assim por diante. Cada vez, a cor do pixel será rastreada com `cordoponto` e substituída pela média dos 3 valores rgb ([verm verde azul]).

Eis o principal do código:

(Não esqueça de alterar o caminho do arquivo no procedimento!)

```
aprenda pixelar :lista
# devolve a média aritmética dos 3 números [r g b]
atr "r pri :lista
atr "lista semprimeiro :lista
atr "g primeiro :lista
atr "lista semprimeiro :lista
atr "b pri :lista
atr "b arredonde (:r+:g+:b)/3
saída sn lista :b :b :b
fim
```

```
aprenda acinzentar :c
se :y=-100 [pare]
se :c=100 [atr "c 0 atr "y :y-1]
# Atribuímos ao lápis a "cor média" do pixel seguinte
mudecl pixelar cordoponto lista :c :y
# Mudamos o ponto para tom de cinza
ponto lista :c :y
acinzentar :c+1
fim
```

```
aprenda transformar
# Note que você deve alterar o endereço de sua imagem (transform.png) logo após carregueimagem
# Por exemplo: carregueimagem [c:\my_images\transform.png]
# XLogo só trabalha com imagens no formato png e jpg
limpedesenho dt carregueimagem [/home/alex/transform.png] atr "y 0 acinzentar 0
fim
```

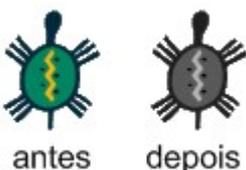


Figura 21: xLogo retoca imagens...

E Se Quisermos um Negativo?

Para fazer o negativo de uma imagem, você deve usar o mesmo processo anterior, exceto que, em vez de usar a média dos números r g b, você deve substituir os números subtraindo 255. Por exemplo, Se um ponto (pixel) tiver a cor [2 100 200], ela será trocada pela cor [253 155 55].

Somente o procedimento pixelar deve ser modificado no programa anterior:

```
aprenda acinzentar :c
# 121 deve ser substituído pela largura da imagem que usar
se :y=-121 [pare]
# 149 deve ser substituído pela altura da imagem que usar
se :c=149 [atr "c 0 atr "y :y-1]
mudecl pixelar cordoponto lista :c :y
ponto lista :c :y
acinzentar :c+1
fim
```

```
aprenda transformar
# Você deverá escrever corretamente o caminho de sua imagem
# Eg: carregueimagem [c:\my_images\nte.png]
dt ld carregueimagem [/home/meulogin/nte.png] atr "y 0 acinzentar 0
fim
```

```
aprenda pixelar :lista
atr "r pri :lista
atr "lista sp :lista
atr "g pri :lista
atr "lista sp :lista
atr "b pri :lista
saída sn lista 255-:r 255-:g 255-:b
fim
```



antes

depois

Figura

Figura 22: xLogo se fazendo de Gimp...
(pretencioso, não? :-))

Uso de Listas

Um bom exemplo de uso de listas (agradecimento a Olivier SC)

Espero que aprecie esse procedimento maravilhoso:

```
aprenda invertep :p
se évazio? :p [saída "]
saída pal ult :p invertep su :p
fim
```

```
aprenda palindromo :p
se sãoiguais? :p invertep :p [saída "verd] [saída "falso]
fim
```

```
aprenda palin :n
se palindromo :n [mo :n pare]
mostre sn sn sn sn :n "mais invertep :n [é igual a] soma :n invertep :n
palin :n + invertep :n
fim
```

```
palin 78
78 mais 87 é igual a 165
165 mais 561 é igual a 726
726 mais 627 é igual a 1353
1353 mais 3531 é igual a 4884
4884
```

Uma Linda Rosácea

```
aprenda rosacea
repita 6 [ repita 60 [pf 2 pd 1] pd 60 repita 120 [pf 2 pd 1] pd 60]
fim

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 mudecl 5 dt rosacea un mudepos [-300 0] ul mudedireção 0 lindarosacea
```

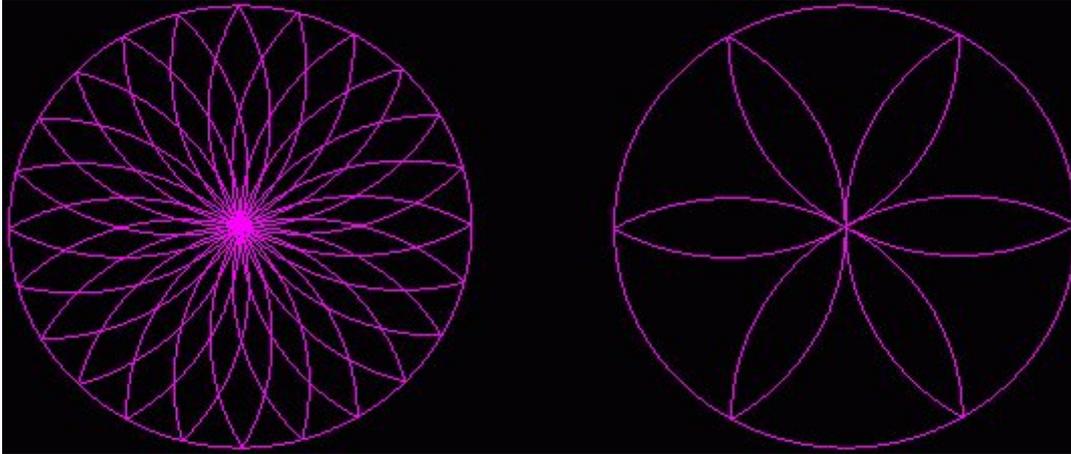


Figura 23: Melhor do que usar um compasso!

Instalar, Desinstalar e Atualizar

Instalar

Para instalar o xLogo é necessário que você tenha instalado em seu micro o "Java Runtime Environment" (JRE) produzido pela Sun.

IMPORTANTE: Este 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 você tem o java instalado e em funcionamento é ir no menu iniciar, clicar em executar, escrever javaw e dar OK.

Se nenhuma mensagem de erro aparecer é porque está OK.

Para saber mais detalhes sobre como instalar o xLogo em Linux ou Windows, leia instruções em

projetologo.webs.com/xlogo.html

Desinstalar

Para desinstalar XLogo, 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.

Atualizar

Confira a última versão e correção de "bugs" no sítio do xLogo com frequência (<http://xlogo.tuxfamily.org>).

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

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

Lista de Primitivas

Apêndice A: 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
atençãoat, atat	fixetortue, ffortue	pontortuga, pportuga	setturtle, sturtle	setzeschildkroete, ssk
atribua, atr	donne	haz	globalmake, make	setze
atrlocal	donnelocale	hazlocal	localmake	lokalsetze
baixenariz	pique	cabeceabajo	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
comandoexterno				
comlimite	enr	modovuelta	wrap	rspr
contador	debuttemps	crono	countdown	zr
conte	compte	cuenta	count	zaehle
conteúdo	contenu	contenido	contents	inhalt
contavezes, cv	compteur	cuentarepite, contador	repcount	wiederholungszahl
cordofundo, cdf	couleurfond, cf	colorpapel	screencolor, sc	bildfarbe, bf
cordolápis, cdl	couleurcrayon, cc	colorlapiz, cl	pencolor, pc	stifffarbe, sf
cordoponto, cdp	trouvecouleur, tc	encuentracolor, ec	findcolor, fc	findefarbe, ff
cordotexto	couleurtexte	colortexto	textcolor	farbtext
data	date	fecha	date	datum
deletesequência, 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
direçãopara, dçpara	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
éanterior?, eant?	precede?	antes?, anterior?	before?, beforep	vorher?
éinteiro?	entier?	entero?	integer?	ganzzahl?
edite	ed	ed	ed	ed
editetudo	edtout	edtodo	edall	edalles
eixo	axes	ejes	axis	achsen

eiox	axex	ejex	xaxis	xachse
eixoy	axey	ejey	yaxis	yachse
eixo?	axes?	ejes?	axis?	achsen?
eiox?	axex?	ejex?	xaxis?	xachse?
eixoy?	axey?	ejey?	yaxis?	yachse?
elemento, elem	item	elemento	item	element
elimine	effacenom	borra	erase	vergiss
eliminetudo, et	effacenomns, 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 , number?	zahl?
enquanto	tantque	mientras	while	solange
envietcp	envoietcp	enviatcp	sendtcp	sendetcp
épalavra?, epal?	mot?	palabra?	word?, wordp	wort?
éprimitiva?, eprim?	primitive?, prim?	primitiva?, prim?	primitive?, prim?	grundwort?, gw?
éprocedimento?, eproc?	procedure?, proc?	procedimiento?, proc?	procedure?, proc?	prozedur?, proz?
escolhe	choix	elige	pick	nimmwas
escondetat, dt	cachetortue, ct	ocultatortuga, ot	hideturtle, ht	versteckeschildkroete
escreva, esc	tape	tipea	write	schreibe
escrevalinha, esfx	ecrisligneflux	escribelineafujo	writelineflow	schreibezeilfluss
escutetcp	ecoutetcp	escuchatcp	listentcp	wartetcp
espere	attends	espera	wait	warte
espessuradolápis, el	taillecrayon, tc	grosorlapiz, gl	penwidth, pw	
étecla?	touche?	tecla?	key?	taste?
éuselápis?	baissecrayon?	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	fermeflux	cierraflujo	closeflow	schliessefluss?
fim	fin	fin	end	ende
fimarq?, fimfluxo?	finflux?	finflujo?	endflow?	endefluss?
fimcontador?	fintemps?	fincrono?	endcountdown?	ezr?
fonte	taillepolic	fuelle	fontsize	schrfa
fontetexto, ftexto	tpt, taillepolicetexte	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
gardeimagem	sauveimage	guardaimagen	saveimage	
igapague	guienleve	eliminaigu	guiremove	entfernegui
igbotão	guibouton	botonigu	guibutton	guitaste
igdesenhe	guidessine	dibujaigu	guidraw	zeichnegui
igfaz	guiaction	accionigu	guiaction	guiaktion
iglocal	guiposition	posicionigu	guilocation	guiposition
igmenu	guimenu	menuigu	guimenu	guimenu

imts	imts	imts	poall	zga
índicesequência, indseq	indexsequence, indseq	índicesecuencia, indsec	indexsequence, indseq	indexfolge, indfol
inicielinha	lignedef	empiezaalínea	linestart	linienanfang
iniciepolígono	polydef	empiezapolígono	polystart	vieleckanfang
inicieponto	pointdef	empiezapunto	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	agregalineafujo	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	leelineafujo	readlineflow	lesezeilefluss
leiamouse	lissouris	leeraton	readmouse	liesmaus
levantenariz	cabre	cabeceaarriba	uppitch	rauf
limpedesenho, ld	videecran, ve	borrapantalla, 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
listear	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	druckezeile, dz
mostrearquivos, marq	catalogue, cat	catalogo, cat	files	zeigedateien, zd
mostretat, at	montretortue, mt	muestratortuga, mt	showturtle, st	zeigeschildkroete, zs
mudecordofundo, mudecf	fixecouleurfondgraphique, fcfg	poncolorpapel, poncp	setscreencolor, setsc	setzebildfarbe, sbf
mudecordolápis, mudecl	fixecouleurcrayon, fcc	poncolorlapiz, poncl	setpencolor, setpc	setzestiffarbe, ssf
mudecordotexto, mudect	fixecouleurtexte, fct	poncolortexto, pctexto	setttextcolor, stc	setzeschriftfarbe, sschrffar
mudedireção, mudedç	fixecap	ponrumbo, ponr	setheading	aufkurs, ak
mudediretório, muedir	fixerepertoire, cd	pondirectorio, cambiadirectorio	setdirectory, changedirectory	setzeordner, wechsleordner
mudeespessuradolápis, mudeel	fixetaillecrayon, ftc	pongrosor	setpenwidth, setpw	stiftdicke, sd
mudefonte, mundef	ftp, fixetaillepolice	ponfuenta, pf	setfontsize, setfs	setzeschriftart, sschrfa
mudefontetexto,	ftpt,	ponfuentetexto,	setttextsize,	setzeschriftarttext,

mudeft	fixetaillepolicetexte	pft	sts	sschrfat
mudeíndice sequência , mudeindseq	fixeindexsequence findseq	ponindice secuencia pindsec	seteindex sequence sindseq	findefolge, findfol
mude instrumento , mudeinstr	fixeinstrument, finstr	pon instrumento pinstr	set instrument sinstr	setze instrument sinstr
mude máximodetartarugas , mudemaxtat	fixemaxtortues, fmt	pon maximastortugas , pmt	setturtles max , stm	setzeschildkröten max
mude nomefonte mudenf	fixenompolice fnp	pon nombrefuente pnf	setfont name setfn	setzesch riftname sschrfna
mude nomefontetexto mudenft	fixenompolicetexte, fnpt	pon nombrefuentetexto , pnft	settext name , stn	setzesch riftnametext , sschrfnat
mude pontadolápis mudepl	fixeformecrayon	pon formalapiz	setpens shape	setzest iff form
mude pos	fpos	pon pos	set pos	spos
mude precisão	fixedecimales	pon decimales	set digits	setze ziffern
mude qualidadedaimagem , mudeqi	fixe qualitedessin	pon calidaddibujo	setdrawing quality	s zeichnung squalität
mude roupa	fforme	p forma	set shape	s form
mude tamanhodajanela	fixetailledessin	pon tamañopantalla	sets creensize	
mude x	fixex	pon x	set x	s x
mude xy	fixexy	pon xy	set xy	s xy
mude y	fixey	pon y	set y	s y
não	non	no	not	nicht
nomefonte	nompolice, np	nombrefuente, nf	fontname	sch rf na
nomefontetexto , nft	nompolicetexte, npt	nombrefuentetexto, nft	textname, tn	sch riftart name, sch rf ana
objeto	chose	objeto	thing	wert
ou	ou	o	or	oder
palavra , pal	mot	palabra	word	wort
para	repetepour	repite para	for	fuer
paradireita , pd	tourned roite , td	giradere cha , gd	right, rt	rechts, re
paraesquerda , pe	tourne gauche , tg	giraiz quierda , 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	detiene animacion	stop anim	stoppe animation
paremp3	stopmp3	detien emp3	mp3 stop	mp3 stoppe
paretudo	stoptout	detien etodo	stop all	hoerauf
perspectiva	perspective	perspectiva	3d	perspektiv isch
pi	pi	pi	pi	pi
pinte	remplis	rellena	fill	fu elle
pintezona	remplis zone	rellenazona	fillzone	form fu ellen
ponto	point	punto	dot	punkt
pos	pos	pos	pos	pos
posmouse	possouris	pos raton	pos mouse	pos maus
potência	puissance	potencia	power	pot
precisão				
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	balanceaizquierda	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	detienecuadrícula	stopgrid	stoppegitter
semlimite	fen	modoventana	window	fen
semprimeiro, sp	saufpremier, sp	menosprimero, mp	butfirst, bf	ohneerstes, oe
semtartaruga , semtat	tuetortue	eliminartortuga	killturtle	loeschschildkroete
semúltimo, su	saufdernier, sd	menosultimo, mu	butlast, bl	ohneletztes, ol
sentença, sn	phrase, ph	frase, fr	sentence, se	satz
sequência, seq	sequence, seq	secuencia, sec	sequence, seq	folge, fol
soma	somme	suma	sum	plus
sorteie	hasard	azar	ran	zuf
tamanhodajanela , tamjan	tailledessin	tamaño pantalla, tpant	screen size	bildgrösse, bildgrosesse
tamanhojanela , tj	taillefenetre, tf	tamañoventana, tv	zone size	.
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
terminepolígono	polyfin	finpolígono	polyend	vieleckende
termineponto	pointfin	finpunto	pointend	punkteende
texto	texte		text	
toque	joue	tocamusica	play	Spiele
toquemp3	jouemp3	escuchamp3	mp3play	mp3spiele
último, ult	der, dernier	ultimo	last	letztes
useborracha , ub	gomme, go	goma, go	penerase, pe	radiere, rd
uselápis , 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

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

Lista de figuras

Apêndice B: Lista de figuras nesse manual

<p>figura 1: A janela principal do xLogo</p> <p>figura 2: A janela do Editor de Procedimentos</p> <p>figura 3: Técnica da Perspectiva</p> <p>figuras 4a, 4b e 4.c: Analogias com Avião</p> <p>figura 5: Quadrados em Perspectiva</p> <p>figura 6: Um Cubo (3D)</p> <p>figura 7: Uma figura interessante (3D)</p> <p>figuras 8a 8b: Janelas de diálogo para luz e neblina</p> <p>figura 9: Exemplo com neblina</p> <p>figura 10: Um desenho a ser pintado</p> <p>figura 11: Uso da primitiva pinte</p> <p>figura 12: Uso da primitiva pintezone</p>	<p>figura 13: Arco-em-Logo</p> <p>figura 14: Uma partitura</p> <p>figura 15: Uso do mouse</p> <p>figura 16: Uma rede simples</p> <p>figura 17: Casas</p> <p>figura 18: Retângulo</p> <p>figura 19: Um fractal (o floco de neve)</p> <p>figura 20: xLogo é show de bola!</p> <p>figura 21: xLogo retoca imagens...</p> <p>figura 22: xLogo se fazendo de Gimp...</p> <p>figura 23: Melhor do que usar um compasso!</p>
--	---

Sobre esse Documento

XLOGO: Manual do Usuário

Essa documentação foi originalmente gerada usando 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

Esse arquivo é mantido com BrOffice.

Tradução para o inglês iniciada por Loïc em 2005-fev-15

Tradução para o português do Brasil por Alexandre R Soares iniciada em 2005-abr-04

Visite as páginas do xLogo (<http://xlogo.tuxfamily.org>)