

MANUAL DE MSWLOGO version 6.5a



1 HIS	storia de MSW Logo	2
2 Use	o del entorno de MSWLogo	2
3 Cr	ear procedimientos en MSWLogo	3
4 De	finición de variables	6
5 Lis	tado de primitivas de MSWLogo	8
5.1	Primitivas para dibujar	8
5.2	Primitivas para escribir, editar y trabajar con ficheros	11
5.3	Primitivas para trabajar con controladoras	15
	Salidas digitales	
	Entradas digitales	
	Entradas Analógicas	
	Salidas Analógicas	
5.4	Primitivas de sonido	21
5.5	Primitivas para crear condicionales y bucles	23
5.6	Primitivas para dibujar ventanas	24
5.7	Primitivas para trabajar con imágenes	33
5.8	Primitivas para salir de MSWLogo	35
6 Bo	tones de la ventana de trabajo	35
7 Co	mentarios	37
8 Gu	ardar un fichero de programación	37
9 Ca	rgar un fichero de programación	38
10 Bib	oliografía	39



1.- Historia de MSWLogo

El lenguaje Logo fue creado en la década de los 60 por Seymour Papert para que los niños aprendieran ideas matemáticas programando con este lenguaje. En las siguientes décadas él y su equipo continuaron con su desarrollo y tratando de crear un programa con un entorno gráfico que fuera a la vez potente y fácil de usar.

En 1994 MSWLogo fue modificado para permitir el control por ordenador a través de los puertos serie y paralelo.

Todas las versiones del software Logo para los sistemas operativos Unix, MS-Dos, Macintosh y Windows desarrolladas inicialmente por la Universidad de California (Berkeley), son gratuitas y pueden ser copiadas sin ningún tipo de restricción por instituciones educativas. De hecho MSWLogo es la versión libre del lenguaje Logo, que se distribuye bajo la licencia pública general (GPL) del proyecto GNU de la "Free Software Foundation".

El símbolo de Logo es una tortuga robotizada que se mueve bajo el control de un ordenador dibujando a medida que se desplaza por la pantalla. En algunas versiones de este lenguaje la tortuga ha evolucionado hasta convertirse en otro tipo de objetos. Por ejemplo, en MSWLogo se ha convertido en un triángulo.

MSWLogo se puede utilizar para múltiples utilidades: para dibujar, para crear sonidos, para manipular una controladora a través del ordenador, etc.

La versión que se intentará explicar en esta documentación es la versión que presenta el Centro Nacional de Información y Comunicación Educativa (CNICE), que es una versión del MSWLogo 6.5a traducida al castellano.

2.- Uso del entorno de MSWLogo

MSWLogo es un lenguaje interpretado, es decir, las órdenes introducidas por el usuario son interpretadas por el ordenador y ejecutadas inmediatamente por orden secuencial. En cambio, los programas "compilados" son convertidos primero a código máquina antes de que cualquier parte del programa pueda empezar a funcionar, como por ejemplo Borland C++.



La pantalla de MSWLogo se divide en 2 partes:

 La pantalla principal: En esta ventana es donde se encuentra la tortuga, mediante la ayuda de ésta se dibujan diferentes figuras, se cargan imágenes, etc. Esta ventana está etiquetada con el titulo de "Pantalla de MSWLogo".

En la parte superior de esta ventana se encuentran diferentes menús para la realización de diversas tareas.

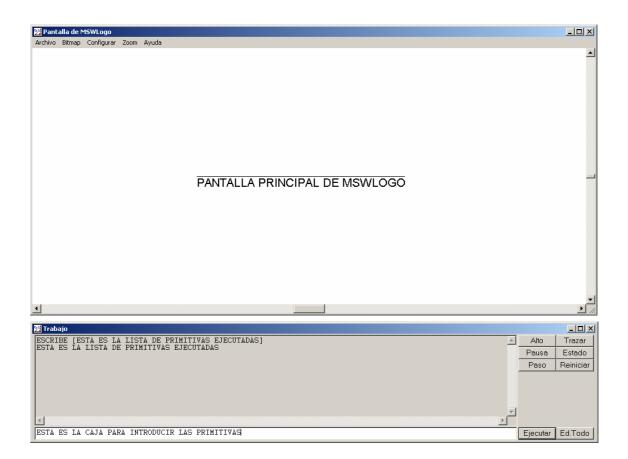


A continuación se muestra una pequeña descripción de las tareas que pueden realizar cada uno de los menús que se muestran en la imagen superior.

- *Archivo:* permite guardar, abrir, cerrar y modificar entre otras cosas los archivos de programación de MSWLogo.
- *Bitmap:* permite poder trabajar con imágenes con extensiones bmp.
- *Configurar:* permite cambiar el color del lápiz, el grosor del lápiz, el tipo de letra, etc.
- *Zoom:* permite acercarse o alejarse de la imagen que se muestra en esta pantalla.
- *Ayuda:* vienen los diferentes archivos de ayuda necesarios para la total comprensión del entorno de MSWLogo.
- **2.** La ventana de trabajo o de comandos: Esta ventana a su vez se divide a su vez en 3 partes diferentes:
 - A) La caja de entrada de datos o primitivas: Se pueden usar mayúsculas o minúsculas para las primitivas puesto que Logo no hace diferencia unas de otras. Las primitivas se ejecutan después de ser introducidas en la ventana de trabajo y de pulsar la tecla ENTER o de hacer clic en el botón Ejecutar.
 - B) Lista de comandos o primitivas ejecutadas: Cada primitiva se graba en una lista de órdenes y comandos que se muestran en la ventana superior a la de entrada de datos.



C) Botones de comandos: Estos botones se encuentran a la derecha de la ventana de trabajo, éstos realizarán diferentes acciones que se explicarán más adelante en este manual.



3.- Creación de procedimientos

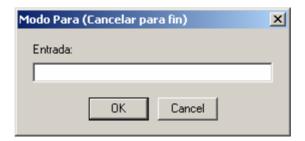
Un procedimiento es una función que ejecuta una serie de líneas de instrucciones, define variables, asigna valor a variables, etc. Para poder crear un procedimiento en MSWLogo hay que utilizar la palabra reservada para seguida del nombre que se vaya a poner al procedimiento y de los parámetros que tenga éste, si es que los tiene. Después se pondrá la definición del procedimiento con toda la lista de instrucciones, para finalizar el procedimiento se utiliza la palabra reservada fin.

```
PARA nombre_del_procedimiento [lista de parámetros]
Lista de instrucciones
.
.
FIN
```



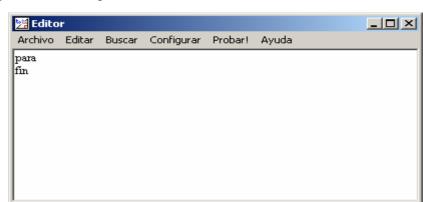
A la hora de crear procedimientos, existen 3 maneras distintas de realizarlos. A continuación las exponemos:

En la ventana de trabajo, hay que escribir en la caja de entrada de datos
o primitivas la siguiente instrucción para nombre [lista de
parámetros], luego se ha de pulsar la tecla ENTER o el botón Ejecutar.
Una vez se ha realizado la anterior acción aparecerá la siguiente ventana.



En esta ventana se irán introduciendo una a una las sentencias o instrucciones de las que va a constar el procedimiento, según se vaya poniendo cada instrucción se irá pulsando el botón *OK* para ir añadiéndolas al procedimiento. Para terminar con la definición del procedimiento se introducirá la primitiva *FIN*.

2. Mediante la ventana de edición, que es una ventana en la que se irán poniendo las instrucciones, previamente tenemos que poner el nombre al procedimiento después de la palabra para. Para poder acceder a esta ventana se utiliza las primitivas editatodo o editatodoprocedimientos, también se puede acceder a esta ventana a través del menú archivo> Editar. Además de definir procedimientos, en la ventana de Edición se pueden definir variables, asignar valor a dichas variables, escribir comentarios, etc. Cuando se ha acabado de definir el procedimiento se guardará en la opción del menú Archivo → Guardar.





3. En un archivo con extensión "*lgo*", dentro de este archivo se pueden definir uno o varios procedimientos mediante las palabras reservadas para y fin. Posteriormente lo que se tiene que hacer es cargar en el MSWLogo el archivo "*lgo*", para lo cual iremos a la opción del menú *Archivo* → *Cargar*.

Una cosa bastante importante de los procedimientos en MSWLogo es la recursividad, es decir un procedimiento se puede llamar a si mismo. Esto es bastante importante ya que se utilizará bastante en la programación de los diferentes proyectos para poder crear pequeños bucles.

También como en otros entornos de programación un procedimiento puede llamar a otro procedimiento, lo que hará esto es realizar las instrucciones que tiene definidas el procedimiento al que se llama.

```
PARA nombre_del_procedimiento1 [lista de parámetros]
Lista de instrucciones
.
.
.
.
FIN

PARA nombre_del_procedimiento2 [lista de parámetros]
Lista de instrucciones
.
.
.
. nombre_del_procedimiento1

FIN
```



4.- Definición de variables

Las variables son una parte fundamental del mundo de la programación, ya que nos sirven para guardar cualquier tipo de información en un área reservada de la memoria del ordenador. El área reservada puede tener una longitud variable o fija, a continuación explicamos los 2 conceptos:

- *Fija*: el tamaño de la variable no cambia a lo largo de la ejecución del programa. Prácticamente todas las variables tienen una longitud fija.
- Variable: el tamaño de la variable va cambiando a lo largo de la ejecución del programa.

Para definir una variable en MSWLogo se utiliza la palabra reservada haz seguida de una apertura de comillas », tras las cuales se pone el nombre que se le vaya a dar a la variable.

```
Haz "nombre variable valor
```

A una variable se la puede asignar un valor, este valor puede ir cambiando a lo largo de la ejecución de un programa. El valor depende del tipo de dato que se quiera guardar en la variable. Ésta puede guardar los siguientes tipos de datos, son:

- Entero
- Lógico
- Carácter
- Cadena de texto

Para asignar un valor a una variable lo único que hay que hacer es poner tras el nombre de la variable el valor que se le quiera dar ya sea un número o una cadena de texto. Así si por ejemplo se quiere asignar un valor de una letra o una palabra, éstas irán precedidas de comillas ("), pero si se quiere asignar como valor una cadena de texto el valor se pone entre corchetes []. A continuación mostraremos varios ejemplos.

```
Haz "nombre_variable 5

Haz "nombre_variable "a

Haz "nombre variable "verdadero
```



Haz "nombre variable [Esto es un ejemplo]

Para poder acceder al valor de una variable se debe anteponer al nombre de la variable dos puntos ":". Así por ejemplo mediante el uso de otra primitiva nos muestra el valor en pantalla de la primitiva.

```
Haz "nombre_variable 5
muestra :nombre_variable
```

En pantalla aparecerá el número 5. Más adelante se explicará el uso de la primitiva "muestra".

También hay otra palabra reservada para poder crear variables y asignarles un valor, la palabra a utilizar es haz2, lo único que cambia con respecto a haz es el orden en la declaración de la variable. Así una variable con un valor asignado se declararía de la siguiente forma:

```
Haz2 valor "nombre_variable
```

Ej: Haz2 5 "nombre_variable

5.- Listado de primitivas seleccionadas de MSWLogo

En este punto se van a comentar las primitivas más utilizadas en el entorno de programación de MSWLogo mediante una sencilla descripción de las funciones que realiza y un pequeño ejemplo de programación.

5.1.- Primitivas para dibujar

PRIMITIVA	MÉTODO	DESCRIPCIÓN	EJEMPLO
	ABREVIADO		
AVANZA unidades	AV	La tortuga avanza el número especificado de unidades.	AVANZA 100 6 AV 100
RETROCEDE unidades	RE	La tortuga retrocede el	RETROCEDE 100 ó

			RE 100
		unidades.	
GIRADERECHA ángulo	D	La tortuga gira en el	GIRADERECHA 90 ó
VI	IRA	sentido de las agujas del	GD 90 6
		reloj el número de grados	VIRA 90
		especificado.	Gira la tortuga 90°
			hacia la derecha
GIRAIZQUIERDA GI	T	La tortuga gira en sentido	GIRAIZQUIERDA 90 ó
ángulo		contrario a las agujas del	GI 90
		reloj el ángulo	Gira la tortuga 90°
		especificado.	hacia la izquierda
PONLAPIZ PI	LA	Pone el lápiz BAJADO y el	PONLAPIZ Ó
		modo PINTA (PINTA es el	PLA
		modo normal de la tortuga	
		para dibujar).	
GOMA GO)	Pone el lápiz BAJADO y el	GOMA Ó
		modo a BORRA (la tortuga va	GO
		borrando por donde se	
		mueve).	
SUBELAPIZ SI	L	Pone el lápiz LEVANTADO,	SUBELAPIZ Ó
		sin cambiar su modo.	SL
BAJALAPIZ BI	L	Pone el lápiz BAJADO, sin	BAJALAPIZ Ó
		cambiar su modo.	BL
CENTRO		Devuelve a la tortuga al	CENTRO
		centro de la pantalla sin	
		borrarla. Hay que utilizar	
		la primitiva SUBELAPIZ	
		para evitar que dibuje	
		mientras va hacia el	
		centro.	
PONCOLORLAPIZ [PO	ONCL [Pone el lápiz del color	El color amarillo es:
1]	determinado de acuerdo con:	
		PONCOLORLAPIZ [255 000 000]	PONCOLORLAPIZ [255
		= Rojo	255 000] ó
		PONCOLORLAPIZ [000 255 000]	PNCL[255 255 000]
		= Verde	
		PONCOLORLAPIZ [000 000 255]	
		= Azul	
OCULTATORTUGA OT	T	Oculta la tortuga (el	OT
		triángulo) en la pantalla.	
MUESTRATORTUGA MI	T	Muestra la tortuga (el	MT
		triángulo) en la pantalla.	



HAZTORTUGA tortuga		Coloca la tortuga en la	HAZTORTUGA 0 [0 100
[PosX PosY ángulo		posición que le indicamos,	45 [av 100]]
[instrucciones]]			
[Instructiones]]		hincándole la posición y el ángulo de ubicación, así como las instrucciones que va a realizar	Este ejemplo coloca la tortuga nº 0 en el unidad 0 del eje "X" y 100 del eje "Y",
			después la rota en un ángulo de 45°, y por último avanza ésta 100 unidades en este sentido
BORRAPANTALLA	BP	Borra la pantalla y coloca a la tortuga en el centro.	BORRAPANTALLA Ó BP
LIMPIA		Borra la pantalla, pero mantiene a la tortuga en su posición actual en la pantalla.	LIMPIA
ROTULA [texto]	RO []	Escribe un texto en la dirección en que se encuentra la tortuga. Este texto se muestra en la pantalla de MSWLogo.	ROTULA [Texto de ejemplo] ó RO [Texto de ejemplo]
PONGROSOR [ancho alto]	PONG []	Coloca el ancho del trazado del lápiz y la altura especificados. MSWLogo utiliza solamente el valor de la anchura. Por defecto el valor de la anchura del lápiz es 1.	PONG [50 50] Aumenta el grosor del
PONPOS [X Y]		Coloca la tortuga en las coordenadas que se indican, siendo el primer número las X y el segundo número las Y.	PONPOS [100 100] Coloca la tortuga en la unidad 100 del eje "X" y en la unidad 100 del eje "Y"
PERSPECTIVA		Se añade a la tortuga una tercera dimensión (eje Z) para que ésta se pueda mover por ésta dimensión. Este comando precede a los comandos que vienen debajo de éste.	PERSPECTIVA
PONBALANCEO grados		Rota la tortuga a una nueva posición absoluta según los	



			Daniel Da
			respecto al eje "Y"
		rota sobre el eje Y. Este	
		comando sólo funciona en	
		modo perspectiva.	
PONCABECEO grados		Rota la tortuga a una nueva	PONCABECEO 90
		posición absoluta según los	Rota la tortuga 90°
		grados que se indican. Ésta	respecto al eje "X"
		rota sobre el eje X. Este	
		comando sólo funciona en	
		modo perspectiva.	
BALANCEA grados	BAL	Rota la tortuga hacia la	BALACEA 90 ó
		derecha los grados que se	BAL 90
		indican. Este comando sólo	Rota la tortuga 90°
		funciona en modo	hacia la derecha
		perspectiva.	
BALANCEAIZQUIERDA	CAI	Rota la tortuga hacia la	BALACEAIAQUIERDA 90 ó
grados		izquierda los grados que se	CAI 90
		indican. Este comando sólo	Rota la tortuga 90°
		funciona en modo	hacia la izquierda
		perspectiva.	
BAJANARIZ grados	BAJAN	Desciende la nariz de la	BAJANARIZ 90 ó
BAUANAKIZ GLAGUS	DAUAN		
		tortuga los grados que se	BAJAN 90
		le indiquen. Este comando	Desciende 90° el
		sólo funciona en modo	sentido de la tortuga
		perspectiva.	

5.2.- Primitivas para escribir, editar y trabajar con ficheros

PRIMITIVA	MÉTODO ABREVIADO	DESCRIPCIÓN	EJEMPLO
MUESTRA texto		Escribe la entrada o	MUESTRA "ejemplo
		entradas en la salida	ó
		de escritura que este	MUESTRA [Esto es un
		actualmente	ejemplo]
		(inicialmente es el	
		terminal). El texto	
		si es una sola	
		palabra llevará	
		comillas de apertura	
		", y si es una cadena	
		de texto va entre	
		corchetes [],	

ESCRIBE texto ESCRIBE texto ESCRIBE la entrada o entradas en la salida de escritura que actualmente es té inicialmente es el terminal). Todas las entradas de texto se escriben en una sola línea. El texto si es una sola palabra llevará comillas de apertura ", y si es una cadena de texto va entre corchetes [], no mostrándose los corchetes.	
ESCRIBE texto ES Escribe la entrada o entradas en la salida de escritura que actualmente esté inicialmente es el terminal). Todas las entradas de texto se escriben en una sola línea. El texto si es una sola palabra llevará comillas de apertura ", y si es una cadena de texto va entre corchetes [], no mostrándose los corchetes.	
entradas en la salida de escritura que actualmente esté inicialmente es el terminal). Todas las entradas de texto se escriben en una sola línea. El texto si es una sola palabra llevará comillas de apertura ", y si es una cadena de texto va entre corchetes [], no mostrándose los corchetes.	
de escritura que actualmente esté ejemplo] inicialmente es el terminal). Todas las entradas de texto se escriben en una sola línea. El texto si es una sola palabra llevará comillas de apertura ", y si es una cadena de texto va entre corchetes [], no mostrándose los corchetes.	s un
actualmente esté inicialmente es el terminal). Todas las entradas de texto se escriben en una sola línea. El texto si es una sola palabra llevará comillas de apertura ", y si es una cadena de texto va entre corchetes [], no mostrándose los corchetes.	s un
inicialmente es el terminal). Todas las entradas de texto se escriben en una sola línea. El texto si es una sola palabra llevará comillas de apertura ", y si es una cadena de texto va entre corchetes [], no mostrándose los corchetes.	
inicialmente es el terminal). Todas las entradas de texto se escriben en una sola línea. El texto si es una sola palabra llevará comillas de apertura ", y si es una cadena de texto va entre corchetes [], no mostrándose los corchetes.	
entradas de texto se escriben en una sola línea. El texto si es una sola palabra llevará comillas de apertura ", y si es una cadena de texto va entre corchetes [], no mostrándose los corchetes.	
entradas de texto se escriben en una sola línea. El texto si es una sola palabra llevará comillas de apertura ", y si es una cadena de texto va entre corchetes [], no mostrándose los corchetes.	
escriben en una sola línea. El texto si es un ejemplo] una sola palabra llevará comillas de apertura ", y si es una cadena de texto va entre corchetes [], no mostrándose los corchetes.	
línea. El texto si es una sola palabra llevará comillas de apertura ", y si es una cadena de texto va entre corchetes [], no mostrándose los corchetes.	
una sola palabra llevará comillas de apertura ", y si es una cadena de texto va entre corchetes [], no mostrándose los corchetes.	
llevará comillas de apertura ", y si es una cadena de texto va entre corchetes [], no mostrándose los corchetes.	
apertura ", y si es una cadena de texto va entre corchetes [], no mostrándose los corchetes.	
una cadena de texto va entre corchetes [], no mostrándose los corchetes.	
va entre corchetes [], no mostrándose los corchetes.	
[], no mostrándose los corchetes.	
los corchetes.	
MUESTRAT texto Muestra la entrada o MUESTRAT "ejemp	 lo
entradas similar al ó	
comando "escribe", MUESTRAT [Estc	es
excepto que no un ejemplo]	
muestra el carácter	
de nueva línea al	
final, y las entradas	
múltiples no son	
separadas por líneas.	
EDITAPROCEDIMIENTOS Abre la ventana del Este comando	se
PROCEDIMIENTOS editor que nos explica en el	tema
permite crear o 3 de este manua	1
modificar	
procedimientos,	
variables que estemos	
creando. Su función	
es similar al botón	
"Editar".	
EDITA CONTENIDO EDITATODO Es similar a la Este comando	se
función anterior. Se explica en el	tema
mantiene esta función 3 de este manua	
por compatibilidad	
con versiones	
anteriores de	
MSWLogo.	
EDITA "nombre_proc ED Abre la ventana del Este comando	

		editor con el nombre	explica en el tema
		de procedimiento que	3 de este manual
		se le haya dado.	
EDITAFICHERO		Abre un editor	EDITAFICHERO
"nombre_fich		estándar con el	"c:\\ejemplo.lgo
11amada		fichero que se haya	rotula [Este el
		indicado. En llamada	fichero a editar]
		se pone un conjunto	Abre el fichero
		de instrucciones a	"ejemplo.lgo" que
		ejecutar cuando se	se encuentra en C,
		sale del editor.	y una vez abierto
			muestra en la
			pantalla de MSWLogo
			el texto del
			comando "rotula"
LISTA		Devuelve una lista	LISTA [a b c]
		cuyos componentes son	Crea una lista de 3
		las entradas, que	elementos. Al ver
		pueden ser palabras,	el contenido de la
		listas o matrices.	lista con el
		Tibeas o maerices.	comando:
			MUESTRA LISTA
			[a b c]
LISTA? objeto		Devuelve verdadero si	Con respecto a la
		el objeto de entrada	lista del ejemplo
		es una lista, y	anterior utilizamos
		devuelve falso si no	este comando
		lo es.	 mediante la ayuda
			del comando
			"muestra" para
			poder ver el
			resultado en
			pantalla.
			MUESTRA LISTA?
			[abc]
			Nos devuelve
			verdadero
			Ó
			MUESTRA LISTA?
			[a b]
			Nos devuelve falso
TADA gontonidas			
TAPA contenidos	ma Damodo	1 ^ 1 + -	mana [[
	TAPATODO	Oculta los procedimientos,	TAPA [[encender] []



		variables, y lista de	Oculta para el
		propiedades pasadas	MSWLogo el
		por la entrada. La	procedimiento
		lista de contenidos	"encender"
		ha de ir entre	encender
		corchetes [].	
TAPADO		Devuelve una lista de	Mediante la ayuda
		contenidos con los	del comando
		elementos tapados del	"muestra" nos dice
		entorno de trabajo.	los procedimientos
			que se han ocultado
			a MSWLogo. Con
			respecto al ejemplo
			anterior:
			MUESTRA TAPADO
			Devuelve encender
BORRA contenidos	BO	Borra del entorno de	
BORRA CONLENIAOS	ВО		
		trabajo los	Elimina el
		procedimientos,	procedimiento
		variables y listas de	encender de MSWLogo
		propiedades que se	
		han pasado en la	
		entrada.	
BORRAPROCEDIMIENTOS		Borra todos los	BORRAPROCEDIMIENTOS
		procedimientos que	
		hay en el entorno de	
		trabajo sin tapar	
BORRATEXTO		Limpia el texto que	BORRATEXTO
		hay en el Terminal.	
CARGA		Lee las instrucciones	CARGA
"nombre fichero		del fichero "lgo"	"C:\\ejemplo.lgo
nombre_richero		para a continuación	Carga el fichero
		realizarlas.	"ejemplo.lgo" que
		realizatias.	
			se encuentra en C
GUARDA		Guarda en un fichero	GUARDA
"nombre_fichero		"lgo" las	"C:\\ejemplo.lgo
		definiciones de todos	Guarda en el
		los procedimientos,	fichero
		variables y listas de	"ejemplo.lgo" que
		propiedades sin	se encuentra en C
		tapar.	
BOARCHIVO	BA	Borra el fichero que	BOARCHIVO



se indica, para que	"C:\\ejemplo.lgo
funcione el comando	ó
el fichero ha de	BA "C:\\ejemplo.lgo
estar cerrado.	Borra el fichero
	"ejemplo.lgo" que
	se encuentra en C

5.3.- Primitivas para trabajar con controladoras

El listado de primitivas que se van a exponer a continuación sirve para poder trabajar con las controladoras ENCONOR y CNICE.

Hay que tener en cuenta que estas controladoras no tienen las mismas conexiones, así por ejemplo no tienen el mismo número de entradas analógicas, por ello hay algunas primitivas que no se pueden utilizar. Además la controladora CNICE no tiene salidas analógicas y por tanto las primitivas que para este tipo de salidas no se podrán usar con ella.

SALIDAS DIGITALES (Numeradas del 0 al 7 la controladora CNICE)

PRIMITIVA	DESCRIPCIÓN	EJEMPLO
M1 "I	Hacer girar en un sentido un	M1 "I
	motor conectado a las salidas	
	digitales 0 y 1. Si el actuador	
	es una bombilla, relé o	
	electroimán, simplemente lo	
	activa. Activa la salida	
	digital 0.	
M1 "D	Hacer girar en sentido	M1 "D
	contrario a la primitiva	
	anterior un motor conectado a	
	las salidas digitales 0 y 1. No	
	se diferencia de la primitiva	
	anterior si están conectados	
	otros actuadores como son una	
	bombilla, relé o electroimán,	
	simplemente lo activa. Activa	
	la salida digital 1.	
M1 "P	Desactiva el actuador que está	M1 "P

	conectado a una de las salidas	
	digitales 0 y 1	
M2 "I, M2 "D, M2 "P	Realiza las mismas acciones que	M2 "I, M2 "D, M2 "P
M3 "I, M3 "D, M3 "P	las primitivas anteriores, lo	
M4 "I, M4 "D, M4 "P	único que cambia son las	
, ,	salidas digitales sobre las que	, ,
	actúan.	
	M2 → Salida 2 y 3	
	M3 → Salida 4 y 5	
	M4 → Salida 6 y 7	
М?	Devuelve una lista con el	Esta primitiva se utiliza
	estado de los 4 motores. Los	junto con la primitiva
	estados son: I, D y P.	"muestra", para ver el
		resultado en pantalla.
		MUESTRA M?
		nos devuelve:
		M [I P D I]
M [lista motores]	Permite activar o desactivar	M [I D P P]
	los 4 motores de manera	
	simultáneamente. En la lista de	
	motores se pone el estado que	
	se quiera por cada motor,	encuentien 103 4 motores.
	separados éstos por un espacio.	
	La lista ha de ir entre	
	corchetes [].	
CONECTAR	Activa todas las salidas	CONECTAR
	digitales de la controladora.	
DESCONECTAR	Desactiva todas las salidas	DESCONECTAR
	digitales de la controladora.	
CONECTA numero	Activa la salida digital	CONECTA 1
		Activa la salida digital 0
	(1,2,3,4,5,6,7,8)	
DESCONECTA numero		DECCONECED 0
DESCONECIA Numero	Desactiva la salida digital	
	indicada en :NUM	Desactiva la salida digital 7
	(1,2,3,4,5,6,7,8)	
CONECTADO?	Devuelve una lista de 4	
	elementos (uno por cada motor)	
	cuyos valores son 0 ó 1 según	"muestra", para ver el
	estén activos o no.	resultado en pantalla.
		MUESTRA CONECTADO?
		nos devuelve:
		M [1 0 1 1]
SALIDA dato	Controla en conjunto las ocho	SALIDA 131
		-



		Tables Tables 12.22 2.22 2.22 2.22
	salidas digitales. Envía a las	
	salidas digitales el dato en	0, 1 y 7 (1+2+128)
	binario equivalente al decimal	
	que se ponga en "dato".	
ENVIAOCTETO dato	Procedimiento para la	ENVIAOCTETO 193
	compatibilidad con las	Activa las salidas digitales
	primitivas ENCONOR (realiza la	0, 6 y 7 (1+64+128)
	misma función que la primitiva	
	SALIDA). Envía a las salidas el	
	número decimal que se ponga en	
	"dato".	
APAGA numero salida	Procedimiento para la	APAGA 4
	compatibilidad con las	
	primitivas ENCONOR. Desconecta	
	la salida digital indicada	Humelo 4
	_	
	"numero_salida" que van de 0 al	
	7, dejando el resto de salidas	
	como estén.	
SALIDA?	Devuelve una lista de ocho	Se utiliza junto con la
	elementos con el estado de las	primitiva "muestra" para
	ocho salidas digitales	mostrar el resultado en
	(devuelve un 1 si la salida	pantalla.
	está activada o un 0 si la	MUESTRA SALIDA?
	salida se encuentra	nos devuelve:
	desactivada.	[1 0 0 0 0 0 1 1]
VS?	Devuelve en un número decimal	Se utiliza junto con la
	el valor en decimal de las	primitiva "muestra" para
	salidas digitales que están	mostrar el resultado en
	activadas.	pantalla.
		MUESTRA VS?
		nos devuelve:
		193

ENTRADAS DIGITALES (Numeradas del 0 al 7 en la controladora CNICE)

PRIMITIVA	DESCRIPCIÓN	EJEMPLO
ENTRADA numero_entrada	Procedimiento para la	Se utiliza junto con la
	compatibilidad con las	primitiva "muestra" para
	primitivas proporcionadas	mostrar el resultado en
	por ENCONOR. Devuelve	pantalla.
	VERDADERO si la entrada	MUESTRA ENTRADA 1

	1/ 1/ 1/ 1	
	digital indicada en	
	"numero_entrada" está	
	conectada y devuelve FALSO	Esto ocurriría si la entrada
	si no está conectada.	digital 1 no estuviera
		conectada
VE?	Devuelve un número decimal	Se utiliza junto con la
	que indica que entradas	primitiva "muestra" para
	digitales están activadas.	mostrar el resultado en
		pantalla.
		MUESTRA VE?
		8
		Nos indica que está activada
		la entrada digital nº 3
SD numero_entrada	Devuelve el valor del sensor	Se utiliza junto con la
	digital que se indica en	primitiva "muestra" para
	"numero_entrada", devuelve	mostrar el resultado en
	un 1 si la entrada está	pantalla.
	activa y 0 si la entrada no	MUESTRA SD1
	está activa.	0
	esca acciva.	Nos indica que la entrada
		digital 0 está desactivada
SD?	Devuelve el estado de todas	Se utiliza junto con la
		primitiva "muestra" para
	forma de lista de 0s o 1s.	mostrar el resultado en
		pantalla.
		MUESTRA SD?
		[1 0 0 0 0 0 1 1]
		Nos indica que están
		activadas las entradas
		digitales 0, 6 y 7.
ESPERAON numero_entrada	Deja el programa parado a la	ESPERAON 1
	espera de que se active la	Deja el programa detenido
	entrada digital indicada en	hasta que se active la
	"numero_entrada".	entrada digital 0
ESPERAOFF numero entrada	Deja el programa parado a la	ESPERAOFF 1
	espera de que se desactive	
	la entrada digital indicada	hasta que se desactive la
	en "numero entrada".	entrada digital 0



ENTRADAS ANALÓGICAS (Numeradas de la 1 a la 4 la controladora CNICE)

PRIMITIVA	DESCRIPCIÓN	EJEMPLO
PRIMITIVA LEEANALOGICA dato	Procedimiento para la compatibilidad con las primitivas proporcionadas por la empresa ENCONOR. Lee y escribe en el puerto los datos necesarios para obtener el valor decimal (valor entre 0 y 255) que proporciona la entrada analógica indicada en "dato" (que puede ser 1, 2, 3, o 4). Si consigue leer las entradas, devuelve su valor, para poder obtener el valor en voltios correspondiente al valor decimal devuelto se consigue aplicando la siguiente formula: Voltaje(V) = (N°*5/256) Primitiva valida para la controladora ENCONOR. Devuelve el valor en voltios	Se utiliza junto con la primitiva "muestra" para mostrar el resultado en pantalla. MUESTRA LEEANALOGICA 1 240 Devuelve el valor en tensión en la entrada analógica 1. Para obtener el valor en voltios de esta entrada se aplica la formula anteriormente descrita. El valor es: 4'68 Voltios Se utiliza junto con la primitiva "muestra" para
SAW	conectado a la entrada analógica 1. Primitiva valida tanto para la controladora CNICE como la ENCONOR.	MUESTRA SAV 0 Nos indica que la entrada analógica 1 de Enconor está desactivada Se utiliza junto con la primitiva "muestra" para mostrar el resultado en pantalla.

SAX	Primitiva valida tanto para	Se utiliza junto con la
	la controladora CNICE como la	primitiva "muestra" para
	ENCONOR.	mostrar el resultado en
	Devuelve el valor en voltios	pantalla.
	de la entrada, esta entrada	MUESTRA SAX
	depende de la controladora	0
	que se este utilizando. Será:	Nos indica que la entrada
	-CNICE: entrada analógica 2	analógica 3 de la
	-ENCONOR: entrada analógica 3	controladora Enconor o la
	Enconort. energia unarogra	entrada 2 de la
		controladora CNICE está
		desactivada
SAY	Primitiva valida tanto para	Se utiliza junto con la
	la controladora CNICE como la	primitiva "muestra" para
	ENCONOR.	mostrar el resultado en
	Devuelve el valor en voltios	pantalla.
	de la entrada, esta entrada	MUESTRA SAY
	depende de la controladora	0
	que se este utilizando. Será:	Nos indica que la entrada
	-CNICE: entrada analógica 3	analógica 4 de la
	-ENCONOR: entrada analógica 4	controladora Enconor o la
	Enconort. energia unarogrea 1	entrada 3 de la
		controladora CNICE está
		desactivada
SAZ	Primitiva valida tanto para	Se utiliza junto con la
	la controladora CNICE como la	primitiva "muestra" para
	ENCONOR.	mostrar el resultado en
	Devuelve el valor en voltios	pantalla.
	de la entrada, esta entrada	MUESTRA SAZ
	depende de la controladora	0
	que se este utilizando. Será:	Nos indica que la entrada
	-CNICE: entrada analógica 4	analógica 5 de la
	-ENCONOR: entrada analógica 5	controladora Enconor o la
		entrada 4 de la
		controladora CNICE está
		desactivada
SA?	Devuelve una lista con el	Se utiliza junto con la
	valor en voltios de todos los	primitiva "muestra" para
	sensores analógicos.	mostrar el resultado en
		pantalla.
		MUESTRA SA?
		MUESTRA SA? [0 4.6875 0.234375 0]
		[0 4.6875 0.234375 0]



	analógicas

SALIDAS ANALÓGICAS (Válidas únicamente para la controladora ENCONOR)

PRIMITIVA	DESCRIPCIÓN	EJEMPLO
SALIDAANALOGICA	Escribe los datos necesarios para	SALIDAANALOGICA 1 100
numero_salida valor_salida	poner en la salida analógica que	A la salida analógica 1 se
	se le indica en "numero_salida"	le asigna el valor de
	el valor extensión indicado en	salida 100. Éste valor
	"valor_salida" (varía entre 0 y	corresponde a un nivel de
	255). Para conocer el valor en	tensión que se calcula con
	tensión en la salida analógica se	la fórmula anteriormente
	utiliza la siguiente fórmula:	descrita.
	Voltaje(V) = (valor_salida/23,8)	El valor es: 4'20 Voltios
VOLTAJE numero_salida	Fija en la salida analógica	VOLTAJE 2 2,5
valor_salida	indicada en "numero_salida" el	A la salida analógica 2 se
	valor en tensión especificado en	le asigna un valor de
	"valor_salida" (varía entre 0	tensión de 2´5 voltios
	10,5).	
VOLTAJE?	Devuelve una lista con el valor	Se utiliza junto con la
	en voltios de todas las salidas	primitiva "muestra" para
	analógicas.	mostrar el resultado en
		pantalla.
		MUESTRA VOLTAJE?
		[0 4.5 0 0]
		Nos muestra que la salida
		analógica 2 tiene un valor
		de 4,5 voltios

5.4.- Primitivas de sonido

PRIMITIVA	DESCRIPCIÓN	EJEMPLO
TONO [flujosonido]	Emite un tono de sonido. La	TONO [700 5000]
	expresión "flujosonido" es una	Emite un tono de sonido
	lista de pares compuesta por:	agudo durante un período
	[frecuencia duracion]	de tiempo de 5 segundos.
	Frecuencia 🗲 Hertzios	
	Duración 🗲 Milésimas de segundo	



	El sonido solo se emite por el	
	altavoz del PC.	
SUENATONO frecuencia	Función similar a la anterior,	SUENATONO [1000 5000]
duracion	que se mantiene de versiones	Emite un tono de sonido
	anteriores de MSWLogo.	más agudo que con el
	anteriores de Mawlogo.	comando anterior durante
		un período de tiempo de 5
		segundos.
ENCIENDESONIDO frecuencia	Emite un sonido cuya frecuencia	ENCIENDESONIDO 1000
	coincide con el parámetro que se	Emite un tono agudo de
	le pasa. El sonido continuará	manera continua. Este
	hasta que se detenga el comando	comando no funciona en
	"APAGASONIDO".	sistemas operativos
	Esta primitiva no funcionará en	Windows 2000 y Windows XP.
	Sistemas Operativos con núcleos	
	NT, ya que utiliza la primitiva	
	"escribepuerto".	
APAGASONIDO	Quitará un sonido que se haya	
	puesto con la primitiva	Este comando no funciona
	"ENCIENDESONIDO frecuencia".	en sistemas operativos
	Esta primitiva no funcionará en	Windows 2000 y Windows XP.
	Sistemas Operativos con núcleos	
	NT, ya que utiliza la primitiva	
	"escribepuerto".	
SUENAWAVE ficherowave	Reproduce un fichero de tipo	SUENAWAVE "C:\\ejemplo.wav
opciones	".wav" según las opciones que se	1+8
	indiquen. Las opciones se indican	Reproduce el fichero
	con un número entero.	"ejemplo.wav" que se
	0-> Sincrono que no devuelve el	encuentra ubicado en C. v
	control hasta que se haya	lo reproduce de manera
	terminado	continuada dejando el
	1-> Asincrono que devuelve el	control a MSWLogo
	control inmediatamente mientras	Control a Piswiogo
	el sonido sigue sonando	
	2 -> No utiliza el sonido por	
	defecto si el indicado no puede	
	oírse	
	8 -> Reproduce el sonido de	
	manera continua hasta que se	
	ejecuta otra primitiva de sonido	
	16 -> No para un sonido que ya	
	estaba sonando	
	Las opciones anteriores se pueden	
	combinar.	



5.5.- Primitivas para crear condicionales y bucles

PRIMITIVA	DESCRIPCIÓN	EJEMPLO
SI condicion [instrucciones]	Realiza una instrucción o un conjunto de instrucciones si se cumple la condición expresada.	La variable "entradas" se ha creado para guardar el valor de las entradas
		digitales. SI (:entradas_digitales=1) [M1 "P]
		Se expresa que si la entrada digital 0 está activada, se desactivan
		las salidas digitales 0 y 1. Si no se cumple la condición, no se ejecuta
		nada.
SISINO condicion	Realiza las primeras	
[instrucciones1] [instrucciones2]	instrucciones si la condición expresada se cumple, y realiza el otro grupo de instrucciones sino	
	se cumple la condición.	SISINO (:entradas_digitales=1) [M1 "P] [salida 255]
		Se expresa que si la entrada digital 0 está activada, se desactivan
		las salidas digitales 0 y 1, pero si no está activada, se encienden todas las salidas
		digitales.
HAZ.HASTA [instrucciones]	Repite la lista de instrucciones	La variable "entradas" se
[condicion]	tantas veces hasta que se cumpla	ha creado para guardar el
	la condicion. Primero lee el	
	conjunto de instrucciones, de	
	esta forma se asegura que las instrucciones se ejecutan al	HAZ.HASTA [SI [:entradas digitales=3]
	menos una vez antes de comprobar	[salida 1]
	la condición.	SI [:entradas_digitales=5] [salida 0]]
		[:bucle=1] Lo primero que hace es chequear las

		instrucciones, que son 2
		condiciones de los "SI", y
		después comprueba si se
		cumple la condición del
		"HAZ.HASTA", si ésta se
		cumple no realiza más
		veces las instrucciones
		del "HAZ.HASTA".
HASTA [condicion]	Repite la lista de instrucciones	La variable "entradas" se
[instrucciones]	tanta veces hasta que se cumpla	ha creado para guardar el
	la condición. Si la condición	valor de las entradas
	expresada se cumple no se	digitales.
	ejecutan las instrucciones ni una	HASTA [:bucle=1]
	sola vez.	[SI
		[:entradas_digitales=3]
		[salida 1]
		SI [:entradas_digitales=5]
		[salida 0]]
		Primero comprueba si se
		cumple la condición, si
		 ésta no se cumple evalúa
		las instrucciones del
		"HASTA".
HAZ.MIENTRAS [instrucciones]	Repite la lista de instrucciones	La variable "entradas" co
[condicion]		ha creado para guardar el
recharcient	condición expresada. Primero lee	
	el conjunto de instrucciones, de	
	er conjunce de inscrucerones, de	
	esta forma se asegura que las	
	esta forma se asegura que las	HAZ.MIENTRAS [
	instrucciones se ejecutan al	HAZ.MIENTRAS [SI [:entradas_digitales=3]
	instrucciones se ejecutan al menos una vez antes de comprobar	HAZ.MIENTRAS [SI [:entradas_digitales=3] [salida 1]
	instrucciones se ejecutan al	HAZ.MIENTRAS [SI [:entradas_digitales=3] [salida 1] SI [:entradas_digitales=5]
	instrucciones se ejecutan al menos una vez antes de comprobar	HAZ.MIENTRAS [SI [:entradas_digitales=3] [salida 1] SI [:entradas_digitales=5] [salida 0]]
	instrucciones se ejecutan al menos una vez antes de comprobar	HAZ.MIENTRAS [SI [:entradas_digitales=3] [salida 1] SI [:entradas_digitales=5] [salida 0]] [:bucle=1]
	instrucciones se ejecutan al menos una vez antes de comprobar	HAZ.MIENTRAS [SI [:entradas_digitales=3] [salida 1] SI [:entradas_digitales=5] [salida 0]] [:bucle=1] Lo primero que hace es
	instrucciones se ejecutan al menos una vez antes de comprobar	HAZ.MIENTRAS [SI [:entradas_digitales=3] [salida 1] SI [:entradas_digitales=5] [salida 0]] [:bucle=1] Lo primero que hace es chequear las
	instrucciones se ejecutan al menos una vez antes de comprobar	HAZ.MIENTRAS [SI [:entradas_digitales=3] [salida 1] SI [:entradas_digitales=5] [salida 0]] [:bucle=1] Lo primero que hace es chequear las instrucciones, que son 2
	instrucciones se ejecutan al menos una vez antes de comprobar	HAZ.MIENTRAS [SI [:entradas_digitales=3] [salida 1] SI [:entradas_digitales=5] [salida 0]] [:bucle=1] Lo primero que hace es chequear las instrucciones, que son 2 condiciones de los "SI", y
	instrucciones se ejecutan al menos una vez antes de comprobar	HAZ.MIENTRAS [SI [:entradas_digitales=3] [salida 1] SI [:entradas_digitales=5] [salida 0]] [:bucle=1] Lo primero que hace es chequear las instrucciones, que son 2 condiciones de los "SI", y después comprueba si se
	instrucciones se ejecutan al menos una vez antes de comprobar	HAZ.MIENTRAS [SI [:entradas_digitales=3] [salida 1] SI [:entradas_digitales=5] [salida 0]] [:bucle=1] Lo primero que hace es chequear las instrucciones, que son 2 condiciones de los "SI", y después comprueba si se cumple la condición del
	instrucciones se ejecutan al menos una vez antes de comprobar	HAZ.MIENTRAS [SI [:entradas_digitales=3] [salida 1] SI [:entradas_digitales=5] [salida 0]] [:bucle=1] Lo primero que hace es chequear las instrucciones, que son 2 condiciones de los "SI", y después comprueba si se cumple la condición del "HAZ.MIENTRAS", si ésta no
	instrucciones se ejecutan al menos una vez antes de comprobar	HAZ.MIENTRAS [SI [:entradas_digitales=3] [salida 1] SI [:entradas_digitales=5] [salida 0]] [:bucle=1] Lo primero que hace es chequear las instrucciones, que son 2 condiciones de los "SI", y después comprueba si se cumple la condición del "HAZ.MIENTRAS", si ésta no se cumple no realiza más
	instrucciones se ejecutan al menos una vez antes de comprobar	HAZ.MIENTRAS [SI [:entradas_digitales=3] [salida 1] SI [:entradas_digitales=5] [salida 0]] [:bucle=1] Lo primero que hace es chequear las instrucciones, que son 2 condiciones de los "SI", y después comprueba si se cumple la condición del "HAZ.MIENTRAS", si ésta no se cumple no realiza más veces las instrucciones
	instrucciones se ejecutan al menos una vez antes de comprobar	HAZ.MIENTRAS [SI [:entradas_digitales=3] [salida 1] SI [:entradas_digitales=5] [salida 0]] [:bucle=1] Lo primero que hace es chequear las instrucciones, que son 2 condiciones de los "SI", y después comprueba si se cumple la condición del

[instrucciones]	tanta veces como se de la	ha creado para guardar el
	condición. Si la condición	valor de las entradas
	expresada no se cumple no se	digitales.
	ejecutan las instrucciones ni una	MIENTRAS [:bucle=1]
	sola vez.	[SI
		[:entradas_digitales=3]
		[salida 1]
		SI [:entradas_digitales=5]
		[salida 0]]
		Primero comprueba si se
		cumple la condición, si
		ésta se cumple evalúa las
		instrucciones del
		"MIENTRAS".
SIEMPRE [instrucciones]	Repite la lista de instrucciones	SIEMPRE[escribe [Esto es
	de forma repetida.	un ejemplo]
		Esto es un ejemplo
		Escribirá de manera
		continua estas frases,
		hasta que se detenga el
		proceso mediante la ayuda
		del botón "ALTO".
REPITE numero [instrucciones]	Repite la lista de instrucciones	REPITE 3 [escribe [Esto es
imiting finestreestenes,	tantas veces como se indica en el	
	número.	Esto es un ejemplo
	Traine 10.	Esto es un ejemplo
		Esto es un ejemplo
		Nos muestra el texto 3
		veces, que es el número de
		repeticiones que hemos
		indicado.



5.6.- Primitivas para dibujar ventanas

PRIMITIVA	DESCRIPCIÓN	EJEMPLO
CREAVENTANA "padre "nombre	Crea una ventana en la pantalla	CREAVENTANA "principal "ej
[titulo] pox poxy ancho alto	de MSWLogo.	[Ejemplo de ventana] 120
[configuración]	-"padre" es el nombre principal	150 287 100 []
	de la ventana.	Esto creará una ventana con
	-"nombre" se utiliza para	el nombre de "ej" en la
	identificar la ventana, y este	cual se muestra la etiqueta
	nombre ha de ser único, ya que	de "Ejemplo de ventana".
	quizás sea padre de otra	El nombre del proceso padre
	ventana.	de esta ventana se llama
	-"pox posy" son las posiciones	"principal", si esta
	de las X y las Y	ventana se pusiera dentro
	respectivamente de la esquina	de otra ventana se pondría
	superior izquierda de la nueva	en nombre del "padre" de la
	ventana.	ventana de la cuelga.
	-"ancho alto" de la nueva	
	ventana.	
	-"configuración" es una lista	
	de instrucciones, esta lista	
	puede estar vacía y añadir los	
	controles más tarde.	
BORRAVENTANA "nombre	Borra la ventana que se indica	BORRAVENTANA "principal
	en "nombre".	Borra la ventana padre de
		todas las ventanas,
		borraría también las
		ventanas que colgarán de
		esta ventana padre.
		BORRAVENTANA "ej
		Borra solamente la ventana
		etiquetada con el nombre
		"ej".
CREAGROUPBOX "padre "nombre	Crea una caja dentro de una	CREAGROUPBOX "ej
posx posy ancho alto	ventana en la que se pueden	"ej_groupbox 5 0 274 85
	agrupar botones, botones de	Crea una caja dentro de la
	radio, botones de selección.	ventana con la etiqueta
	-"padre" es el nombre de la	"ej" a la cual se llama
	ventana en la que crea el	"ej_groupbox".
	"groupbox".	
	-"nombre" se utiliza para	

	de eserciates	
	de ser único.	
	-"pox posy" son las posiciones	
	de las X y las Y	
	respectivamente de la esquina	
	superior izquierda de izquierda	
	del nuevo "groupbox.	
	-"ancho alto" del nuevo	
	"groupbox".	
BORRAGROUPBOX "nombre	Borra el groupbox que se indica	BORRAGROUPBOX "ei groupbox
	en "nombre".	Borra la caja etiquetada
		como "ej groupbox".
		_
CREABOTON "padre "nombre posx	Crea un botón dentro de una	
posy ancho alto [11amada]	ventana.	[Botón] 25 10 110 35 []
	-"padre" es el nombre de la	Se crea un botón dentro de
	ventana en la que crea el	la ventana etiquetada con
	botón.	"ej" al cual se etiqueta
	-"nombre" se utiliza para	con el nombre de
	identificar el botón y ha de	"ej_boton", y muestra el
	ser único.	texto de "Botón".
	-"pox posy" son las posiciones	
	de las X y las Y	
	respectivamente de la esquina	
	superior izquierda del nuevo	
	botón.	
	-"ancho alto" del nuevo botón	
	creado.	
	- "llamada" es una pequeña	
	lista de instrucciones o un	
	nombre de un procedimiento a	
	llamar cuando se pulsa el	
	botón.	
ACTUALIZABOTON "nombre [texto]	Sustituye el texto que aparece	ACTUALIZABOTON "ej_boton
	dentro del botón.	[Botón actualizado]
	-"nombre" se utiliza para saber	Cambia el texto del botón
	que "ventana" cuelga.	etiquetado con "ej_boton"
		por la palabra "Escribe" en
		vez de por la palabra
		"Botón actualizado".
POPPA POWON Washing	Down at both me and trade.	
BORRABOTON "nombre	Borra el botón que se indica en	_
	"nombre".	Borra el botón etiquetado
		como "ej_boton".
CREABOTONRADIO "padre "grupo	Crea un botón que da al usuario	CREABOTONRADIO "ej
"nombre [etiqueta] posx posy	la posibilidad de selección de	"ej_groupbox "ej_botonradio
ancho alto	un elemento de 2 estados	[Esto es otro ejemplo] 25
L		

	(verdadero o falso). Este tipo	50 120 30
	de botón tiene que estar	Crea un botón de selecció
	asociado con un "groupbox".	dentro del groupbox al qu
	-"padre" es el nombre de la	se etiqueta co.
	ventana en la que crea este	"ej botonradio", el cua
	botón.	tendrá el texto "Esto es u
	-"grupo" es el nombre del	
	"groupbox" al que está	
	asociado.	
	-"nombre" se utiliza para	
	-	
	identificar el botón y ha de	
	ser único.	
	-"etiqueta" se utiliza como	
	etiqueta que aparece en el	
	nuevo botón.	
	-"pox posy" son las posiciones	
	de las X y las Y	
	respectivamente de la esquina	
	superior izquierda del nuevo	
	botón.	
	-"ancho alto" del nuevo botón	
	creado.	
BORRABOTONRADIO "nombre	Borra el botón que se indica en	BORRABOTONRADIO
	"nombre".	"ej botonradio
		 Borra el botón etiquetad
		como "ej_botonradio".
CREACHECKBOX "padre "grupo	Crea un botón que da al usuario	_
	Crea un botón que da al usuario	CREACHECKBOX "e
"nombre [etiqueta] posx posy	la posibilidad de selección de	CREACHECKBOX "e "ej_groupbox "ej_checkbo
"nombre [etiqueta] posx posy	la posibilidad de selección de un elemento. Este tipo de botón	CREACHECKBOX "e "ej_groupbox "ej_checkbo [Texto de ejemplo] 155 1
"nombre [etiqueta] posx posy	la posibilidad de selección de un elemento. Este tipo de botón tiene que estar asociado con un	CREACHECKBOX "ej_groupbox "ej_checkbo" [Texto de ejemplo] 155 1
"nombre [etiqueta] posx posy	la posibilidad de selección de un elemento. Este tipo de botón tiene que estar asociado con un "groupbox".	CREACHECKBOX "e "ej_groupbox "ej_checkbox [Texto de ejemplo] 155 1 110 35 Crea el botón "checkbox
"nombre [etiqueta] posx posy	la posibilidad de selección de un elemento. Este tipo de botón tiene que estar asociado con un "groupbox". -"padre" es el nombre de la	CREACHECKBOX "ej_groupbox "ej_checkbo" [Texto de ejemplo] 155 1 110 35 Crea el botón "checkbox dentro del "groupbox" co
"nombre [etiqueta] posx posy	la posibilidad de selección de un elemento. Este tipo de botón tiene que estar asociado con un "groupbox". -"padre" es el nombre de la ventana en la que crea este	CREACHECKBOX "e "ej_groupbox "ej_checkbo [Texto de ejemplo] 155 1 110 35 Crea el botón "checkbox dentro del "groupbox" co nombre de "ej_checkbox" e
"nombre [etiqueta] posx posy	la posibilidad de selección de un elemento. Este tipo de botón tiene que estar asociado con un "groupbox". -"padre" es el nombre de la ventana en la que crea este botón.	CREACHECKBOX "ej_groupbox "ej_checkbo" [Texto de ejemplo] 155 1 110 35 Crea el botón "checkbox dentro del "groupbox" co nombre de "ej_checkbox" el cual se muestra "Text
"nombre [etiqueta] posx posy	la posibilidad de selección de un elemento. Este tipo de botón tiene que estar asociado con un "groupbox". -"padre" es el nombre de la ventana en la que crea este botón. -"grupo" es el nombre del	CREACHECKBOX "e "ej_groupbox "ej_checkbo [Texto de ejemplo] 155 1 110 35 Crea el botón "checkbox dentro del "groupbox" co nombre de "ej_checkbox" e
"nombre [etiqueta] posx posy	la posibilidad de selección de un elemento. Este tipo de botón tiene que estar asociado con un "groupbox". -"padre" es el nombre de la ventana en la que crea este botón. -"grupo" es el nombre del "groupbox" al que está	CREACHECKBOX "e "ej_groupbox "ej_checkbo [Texto de ejemplo] 155 1 110 35 Crea el botón "checkbox dentro del "groupbox" co nombre de "ej_checkbox" e el cual se muestra "Text
"nombre [etiqueta] posx posy	la posibilidad de selección de un elemento. Este tipo de botón tiene que estar asociado con un "groupbox". -"padre" es el nombre de la ventana en la que crea este botón. -"grupo" es el nombre del "groupbox" al que está asociado.	CREACHECKBOX "e "ej_groupbox "ej_checkbo [Texto de ejemplo] 155 1 110 35 Crea el botón "checkbox dentro del "groupbox" co nombre de "ej_checkbox" e el cual se muestra "Text
"nombre [etiqueta] posx posy	la posibilidad de selección de un elemento. Este tipo de botón tiene que estar asociado con un "groupbox". -"padre" es el nombre de la ventana en la que crea este botón. -"grupo" es el nombre del "groupbox" al que está asociado. -"nombre" se utiliza para	CREACHECKBOX "e "ej_groupbox "ej_checkbo [Texto de ejemplo] 155 1 110 35 Crea el botón "checkbox dentro del "groupbox" co nombre de "ej_checkbox" e el cual se muestra "Text
"nombre [etiqueta] posx posy	la posibilidad de selección de un elemento. Este tipo de botón tiene que estar asociado con un "groupbox". -"padre" es el nombre de la ventana en la que crea este botón. -"grupo" es el nombre del "groupbox" al que está asociado.	CREACHECKBOX "e "ej_groupbox "ej_checkbo [Texto de ejemplo] 155 1 110 35 Crea el botón "checkbox dentro del "groupbox" co nombre de "ej_checkbox" e el cual se muestra "Text
"nombre [etiqueta] posx posy	la posibilidad de selección de un elemento. Este tipo de botón tiene que estar asociado con un "groupbox". -"padre" es el nombre de la ventana en la que crea este botón. -"grupo" es el nombre del "groupbox" al que está asociado. -"nombre" se utiliza para	CREACHECKBOX "e "ej_groupbox "ej_checkbo [Texto de ejemplo] 155 1 110 35 Crea el botón "checkbox dentro del "groupbox" co nombre de "ej_checkbox" e el cual se muestra "Text
"nombre [etiqueta] posx posy	la posibilidad de selección de un elemento. Este tipo de botón tiene que estar asociado con un "groupbox". -"padre" es el nombre de la ventana en la que crea este botón. -"grupo" es el nombre del "groupbox" al que está asociado. -"nombre" se utiliza para identificar el botón y ha de	CREACHECKBOX "e "ej_groupbox "ej_checkbo [Texto de ejemplo] 155 1 110 35 Crea el botón "checkbox dentro del "groupbox" co nombre de "ej_checkbox" e el cual se muestra "Text
CREACHECKBOX "padre "grupo "nombre [etiqueta] posx posy ancho alto	la posibilidad de selección de un elemento. Este tipo de botón tiene que estar asociado con un "groupbox". -"padre" es el nombre de la ventana en la que crea este botón. -"grupo" es el nombre del "groupbox" al que está asociado. -"nombre" se utiliza para identificar el botón y ha de ser único.	CREACHECKBOX "e "ej_groupbox "ej_checkbo [Texto de ejemplo] 155 1 110 35 Crea el botón "checkbox dentro del "groupbox" co nombre de "ej_checkbox" e el cual se muestra "Text
"nombre [etiqueta] posx posy	la posibilidad de selección de un elemento. Este tipo de botón tiene que estar asociado con un "groupbox". -"padre" es el nombre de la ventana en la que crea este botón. -"grupo" es el nombre del "groupbox" al que está asociado. -"nombre" se utiliza para identificar el botón y ha de ser único. -"etiqueta" se utiliza como	CREACHECKBOX "e "ej_groupbox "ej_checkbo [Texto de ejemplo] 155 1 110 35 Crea el botón "checkbox dentro del "groupbox" co nombre de "ej_checkbox" e el cual se muestra "Text
	la posibilidad de selección de un elemento. Este tipo de botón tiene que estar asociado con un "groupbox". -"padre" es el nombre de la ventana en la que crea este botón. -"grupo" es el nombre del "groupbox" al que está asociado. -"nombre" se utiliza para identificar el botón y ha de ser único. -"etiqueta" se utiliza como etiqueta que aparece en el	CREACHECKBOX "e "ej_groupbox "ej_checkbox [Texto de ejemplo] 155 1 110 35 Crea el botón "checkbox dentro del "groupbox" co nombre de "ej_checkbox" e el cual se muestra "Text

	rognostivamento de la camilia	
	respectivamente de la esquina	
	superior izquierda del nuevo	
	botón.	
	-"ancho alto" del nuevo botón	
	creado.	
BORRACHECKBOX "nombre	Borra el botón que se en	BORREACHECKBOX "ej_checkbox
	"nombre".	Borra el botón etiquetado
		como "ej checkbox".
CDTITION No. Jan. No. or Jan.		
CREALISTBOX "padre "nombre	Crea una lista de selección de	_
posx posy ancho alto	diferentes elementos.	155 45 90 20
	-"padre" es el nombre de la	
	ventana en la que crea este	dentro de la ventana
	botón.	etiquetada como "ej", a
	-"nombre" se utiliza para	esta lista se la etiqueta
	identificar la lista y ha de	como "ej_listbox".
	ser único.	
	-"pox posy" son las posiciones	
	de las X y las Y	
	respectivamente de la esquina	
	superior izquierda de la lista.	
	-"ancho alto" de la lista.	
BORRALISTBOX "nombre	Borra la lista de selección que	BORRALISTBOX "ej_listbox
	se indica en "nombre".	Borra la lista de selección
		etiquetada como
		"ej_listbox".
CREACOMBOBOX "padre "nombre	Crea una caja que se utiliza	
CREACOMBOBOX "padre "nombre posx posy ancho alto		CREACOMBOBOX "ej
_	para dar al usuario una	CREACOMBOBOX "ej" "ej_combobox 155 55 90 20
_	para dar al usuario una selección de múltiples	CREACOMBOBOX "ej "ej_combobox 155 55 90 20 Crea una caja de múltiples
_	para dar al usuario una selección de múltiples elementos.	CREACOMBOBOX "ej "ej_combobox 155 55 90 20 Crea una caja de múltiples elementos dentro de la
_	para dar al usuario una selección de múltiples elementos"padre" es el nombre de la	CREACOMBOBOX "ej "ej_combobox 155 55 90 20 Crea una caja de múltiples elementos dentro de la ventana "ej", a la cual se
	para dar al usuario una selección de múltiples elementos"padre" es el nombre de la ventana en la que se crea.	CREACOMBOBOX "ej "ej_combobox 155 55 90 20 Crea una caja de múltiples elementos dentro de la
_	para dar al usuario una selección de múltiples elementos"padre" es el nombre de la ventana en la que se crea"nombre" se utiliza para	CREACOMBOBOX "ej "ej_combobox 155 55 90 20 Crea una caja de múltiples elementos dentro de la ventana "ej", a la cual se
_	para dar al usuario una selección de múltiples elementos. -"padre" es el nombre de la ventana en la que se crea. -"nombre" se utiliza para identificar el botón y ha de	CREACOMBOBOX "ej "ej_combobox 155 55 90 20 Crea una caja de múltiples elementos dentro de la ventana "ej", a la cual se
_	para dar al usuario una selección de múltiples elementos. -"padre" es el nombre de la ventana en la que se crea. -"nombre" se utiliza para identificar el botón y ha de ser único.	CREACOMBOBOX "ej "ej_combobox 155 55 90 20 Crea una caja de múltiples elementos dentro de la ventana "ej", a la cual se
_	para dar al usuario una selección de múltiples elementos. -"padre" es el nombre de la ventana en la que se crea. -"nombre" se utiliza para identificar el botón y ha de	CREACOMBOBOX "ej "ej_combobox 155 55 90 20 Crea una caja de múltiples elementos dentro de la ventana "ej", a la cual se
_	para dar al usuario una selección de múltiples elementos. -"padre" es el nombre de la ventana en la que se crea. -"nombre" se utiliza para identificar el botón y ha de ser único.	CREACOMBOBOX "ej "ej_combobox 155 55 90 20 Crea una caja de múltiples elementos dentro de la ventana "ej", a la cual se
_	para dar al usuario una selección de múltiples elementos. -"padre" es el nombre de la ventana en la que se crea. -"nombre" se utiliza para identificar el botón y ha de ser único. -"pox posy" son las posiciones	CREACOMBOBOX "ej "ej_combobox 155 55 90 20 Crea una caja de múltiples elementos dentro de la ventana "ej", a la cual se
_	para dar al usuario una selección de múltiples elementos. -"padre" es el nombre de la ventana en la que se crea. -"nombre" se utiliza para identificar el botón y ha de ser único. -"pox posy" son las posiciones de las X y las Y	CREACOMBOBOX "ej "ej_combobox 155 55 90 20 Crea una caja de múltiples elementos dentro de la ventana "ej", a la cual se
_	para dar al usuario una selección de múltiples elementos. -"padre" es el nombre de la ventana en la que se crea. -"nombre" se utiliza para identificar el botón y ha de ser único. -"pox posy" son las posiciones de las X y las Y respectivamente de la esquina	CREACOMBOBOX "ej "ej_combobox 155 55 90 20 Crea una caja de múltiples elementos dentro de la ventana "ej", a la cual se
_	para dar al usuario una selección de múltiples elementos. -"padre" es el nombre de la ventana en la que se crea. -"nombre" se utiliza para identificar el botón y ha de ser único. -"pox posy" son las posiciones de las X y las Y respectivamente de la esquina superior izquierda del nuevo botón.	CREACOMBOBOX "ej "ej_combobox 155 55 90 20 Crea una caja de múltiples elementos dentro de la ventana "ej", a la cual se
	para dar al usuario una selección de múltiples elementos. -"padre" es el nombre de la ventana en la que se crea. -"nombre" se utiliza para identificar el botón y ha de ser único. -"pox posy" son las posiciones de las X y las Y respectivamente de la esquina superior izquierda del nuevo botón. -"ancho alto" del nuevo botón	CREACOMBOBOX "ej "ej_combobox 155 55 90 20 Crea una caja de múltiples elementos dentro de la ventana "ej", a la cual se
posx posy ancho alto	para dar al usuario una selección de múltiples elementos. -"padre" es el nombre de la ventana en la que se crea. -"nombre" se utiliza para identificar el botón y ha de ser único. -"pox posy" son las posiciones de las X y las Y respectivamente de la esquina superior izquierda del nuevo botón. -"ancho alto" del nuevo botón creado.	CREACOMBOBOX "ej "ej_combobox 155 55 90 20 Crea una caja de múltiples elementos dentro de la ventana "ej", a la cual se llamará "ejcombobox".
_	para dar al usuario una selección de múltiples elementos. -"padre" es el nombre de la ventana en la que se crea. -"nombre" se utiliza para identificar el botón y ha de ser único. -"pox posy" son las posiciones de las X y las Y respectivamente de la esquina superior izquierda del nuevo botón. -"ancho alto" del nuevo botón creado. Borra la caja de selección de	CREACOMBOBOX "ej "ej_combobox 155 55 90 20 Crea una caja de múltiples elementos dentro de la ventana "ej", a la cual se llamará "ejcombobox". BORRACOMBOBOX "ejcombobox
posx posy ancho alto	para dar al usuario una selección de múltiples elementos. -"padre" es el nombre de la ventana en la que se crea. -"nombre" se utiliza para identificar el botón y ha de ser único. -"pox posy" son las posiciones de las X y las Y respectivamente de la esquina superior izquierda del nuevo botón. -"ancho alto" del nuevo botón creado. Borra la caja de selección de	CREACOMBOBOX "ej "ej_combobox 155 55 90 20 Crea una caja de múltiples elementos dentro de la ventana "ej", a la cual se llamará "ejcombobox".

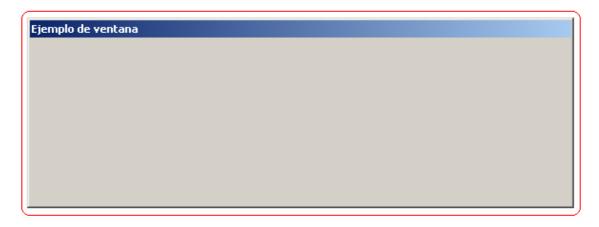


		"ej_combobox".
CREAESTATICO "padre "nombre	Permite mostrar dentro de una	CREAESTATICO "ej
[texto] posx posy ancho alto	ventana un texto. Este texto se	"ej_estatico [ESTATICO] 35
	puede actualizar con la	70 50 10
	primitiva "actualizaestatico".	Crea una ventana de texto
	-"padre" es el nombre de la	dentro de la ventana "ej" a
	ventana en la que se crea.	la cual se llama
	-"nombre" se utiliza para	"ej_estatico". Esta ventana
	identificar el texto que se va	contiene el texto
	a introducir en la ventana y ha	"ESTATICO".
	de ser único.	
	-"pox posy" son las posiciones	
	de las X y las Y	
	respectivamente de la esquina	
	superior izquierda de donde se	
	va a ubicar el texto.	
	-"ancho alto" del texto	
	introducido en la ventana.	
ACTUALIZAESTATICO "nombre	Sustituye el texto que se crea	ACTUALIZAESTATICO
[texto]	con "creaestatico".	"ej_estatico [ACTUALIZADO]
	-"nombre" se utiliza para saber	Actualiza el texto de la
	de que "creaestatico" cuelga.	ventana de texto estática
		de "ESTATICO" a
		"ACTUALIZADO".
BORRAESTATICO "nombre	Borra el botón estático que se	BORRAESTATICO "ej_estatico
	indica en "nombre".	Borra la ventana de texto
		etiquetada como
		"ej estatico".

A continuación vamos a mostrar el resultado en pantalla de los comandos explicados anteriormente en orden progresivo en la creación de ventanas. Dentro de un marco rojo estarán los resultados de los comandos utilizados.



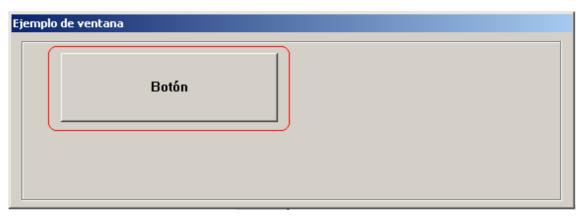
• Crear ventana



• Crear groupbox



• Crear botón





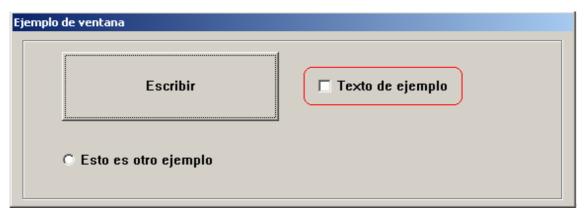
• Actualizar botón



• Crear botón radio

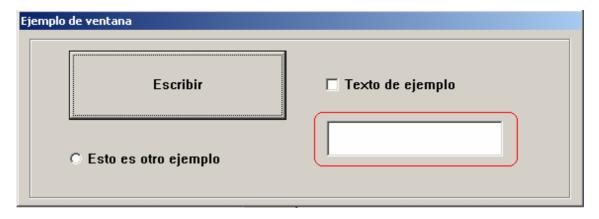


• Crear checkbox

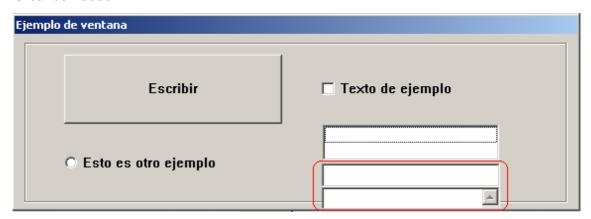




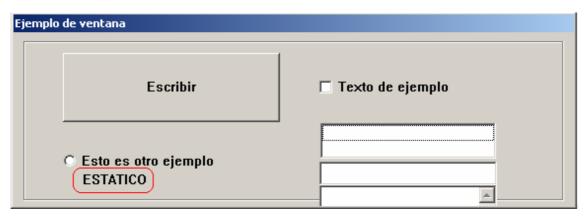
• Crear listbox



• Crear combobox

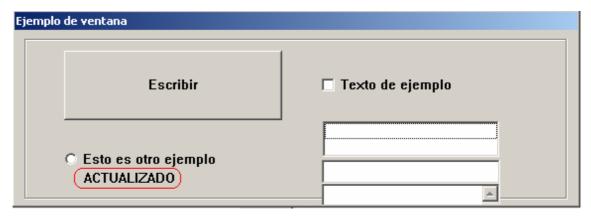


• Crear estático





• Actualizar estático



5.7.- Primitivas para trabajar con imágenes

PRIMITIVA	DESCRIPCIÓN	EJEMPLO
CARGADIB "ruta_imagen.bmp	Carga una imagen ".bmp" que se encuentra donde indica "ruta_imagen.bmp".	CARGADIB "C:\\ejemplo.bmp
CARGADIBTAMAÑO "ruta_imagen.bmp	Devuelve el tamaño de la imagen que se indica en "ruta_imagen.bmp". El tamaño lo devuelve en una lista de 2 números enteros [Ancho Alto].	Se utiliza junto con la primitiva "muestra" para mostrar el resultado en pantalla. MUESTRA "CARGADIBTAMAÑO"C:\\ejemplo.bmp Nos devuelve: [256 256]
GUARDADIB "ruta\nombre_bitmap	Guarda una imagen en formato ".bmp" en la ubicación del ordenador que se especifique.	GUARDADIB "C:\\ejemplo.bmp
CARGAGIF "ruta_imagen.gif	Carga una imagen ".gif" que se encuentra donde indica "ruta_imagen.gif".	CARGAGIF "C:\\ejemplo.gif
GUARDAGIF "ruta\nombre_gif	Guarda una imagen en formato ".gif" en la ubicación del ordenador que se especifique.	GUARDAGIF "C:\\ejemplo.gif



5.8.- Primitivas para salir de MSWLogo

PRIMITIVA	DESCRIPCIÓN	EJEMPLO
ADIOS	Cierra el programa de MSWLogo.	ADIOS

6.- Entorno gráfico de MSWLogo

Como se ha visto al principio del manual de MSWLogo el entorno gráfico de MSWLogo se divide en 2 partes:

- Pantalla de MSWLogo
- Pantalla de Trabajo

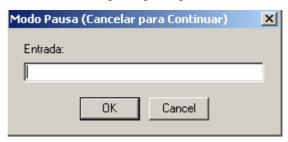
En este punto se van a explicar lo que hacen cada uno de los botones que aparecen en la parte derecha de la ventana de Trabajo de MSWLogo.



- *Alto:* detiene la ejecución del programa que se esta ejecutando sobre MSWLogo.
- *Trazar:* permite ver en el historial de la pantalla de Trabajo las sentencias del programa de MSWLogo que se están ejecutando tiempo real. Se muestra en la imagen de abajo el historial de las sentencias que se están ejecutando.



- *Pausa:* establece una pausa en la ejecución del programa que se esta ejecutando sobre MSWLogo. Nos mostrará una ventana en pantalla en la tendremos que darle a continuar para que siga el desarrollo del programa.



 Estado: muestra en una tabla información sobre el estado del lápiz, la orientación de éste, el color del lápiz, la posición de la tortuga y el estado del núcleo de MSWLogo.



 Paso: ejecuta el programa de MSWLogo paso a paso, es decir, línea a línea del archivo de programación, teniendo que dar consentimiento para que se ejecute la siguiente línea de programación. Nos muestra una ventana similar de la que se muestra a continuación.



- *Reiniciar:* reinicia la ejecución del programa de MSWLogo que se esta ejecutando.



- *Ejecutar:* ejecuta las sentencias de programación de MSWLogo que se escriben en la ventana blanca de la pantalla de Trabajo.
- Editar: abre el cuadro de edición de MSWLogo.

7.- Comentarios

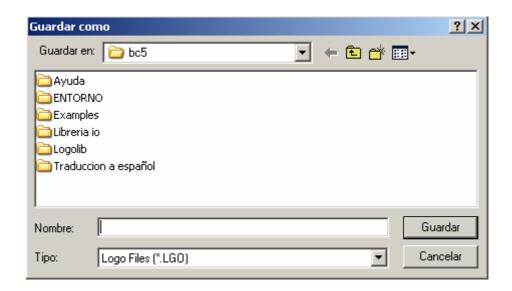
Los comentarios en el lenguaje de MSWLogo han de ir precedidos por un punto y coma ";". A continuación se muestra un ejemplo.

; Esto es un comentario de MSWLogo

8.- Guardar un fichero de programación

Hay 2 maneras diferentes de guardar todo el trabajo que se ha realizado en un sesión de MSWLogo, los procedimientos que hayamos creado con sus variables, sus funciones, etc. A continuación se muestran las 2 maneras diferentes:

- Desde el menú de *Archivo* → *Guardar* o *Archivo* → *Guardar como* podemos guardar lo trabajado con un nombre con extensión "lgo".





- Se utiliza la primitiva guarda "nombfichero.lgo en el que se guarda las definiciones de todos los procedimientos, variables y listas de propiedades sin tapar.

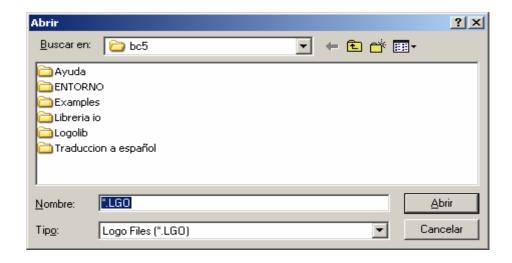


En ambos casos hay que destacar que no se guarda cada procedimiento en un archivo, sino que todo el trabajo definido durante una sesión de MSWLogo se guarda junto en el mismo archivo, excepto aquellas variables o procedimientos que se hayan tapado.

9.- Cargar un fichero de programación

Hay 2 maneras diferentes de cargar un fichero de programación de MSWLogo. A continuación se muestran las 2 maneras diferentes:

- Desde el menú de *Archivo* → *Cargar* se abre una ventana desde donde podemos cargar un archivo de programación de MSWLogo. En esta ventana solo aparecen los archivos con extensión ".lgo". Para que aparezcan todos los archivos se debe seleccionar en *Tipo de archivos*: todos los archivos "*.*".





- Se utiliza la primitiva carga "nombfichero.lgo para que cargue el archivo de programación de MSWLogo especificado. Si el archivo no está dentro de la carpeta "bc5", se tendrá que poner la ruta de acceso a ese fichero.



Es importante destacar que cuando se carga un fichero "lgo" se cargan todos sus procedimientos en el entorno de MSWLogo.

Si no se han guardado el entorno de trabajo de la sesión actual en disco o si en el archivo existen procedimientos ya definidos en la sesión actual, se sobrescribirá. Pero antes de eso MSWLogo emite una ventana de aviso:



10.- Bibliografía

- Guía práctica de MSWLogo en inglés:
 http://www.southwest.com.au/~jfuller/logotut/logo1.htm
- Manuales de Logo en español:
 http://www.southwest.com.au/~jfuller/mswlogo/spanish.zip
 http://www.angeltowns.com/members/mondragon/paradiso/ sect2