MANUAL DE MSWLOGO versión 6.5a



1 His	storia de MSWLogo	2
2 Use	o del entorno de MSWLogo	2
3 Cro	ear procedimientos en MSWLogo	3
4 Det	finición de variables	7
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	14
	Salidas digitales	
	Entradas digitales	
	Entradas Analógicas	
	Salidas Analógicas	
5.4	Primitivas de sonido	20
5.5	Primitivas para crear condicionales y bucles	22
5.6	Primitivas para dibujar ventanas	25
5.7	Primitivas para trabajar con imágenes	33
5.8	Primitivas para salir de MSWLogo	33
6 Ent	torno grafico de MSWLogo	33
7 Co	mentarios	35
8 Gu	ardar un fichero de programación	36
9 Ca	rgar un fichero de programación	37
10 Bit	bliografía	38

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 Instituto Superior de Formación y Recursos en Red para el Profesorado (ISFTIC), 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.



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.

Modo Para (Cancelar para fin)		
Entrada:		
OK Cancel		

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.

Interfaz de control de dispositivos externos por ordenador a través de puerto paralelo

🔀 Editor							
Archivo	Editar	Buscar	Configurar	Probar!	Ayuda		
para fin							

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
```

El ordenador como elemento de control



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 5Haz "nombre_variable "aHaz "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 ó AV 100
RETROCEDE unidades	RE	La tortuga retrocede el	RETROCEDE 100 ó



		número especificado de	RE 100
		unidades.	
GIRADERECHA ángulo	GD	La tortuga gira en el	GIRADERECHA 90 ó
	VIRA	sentido de las agujas	GD 90 ó
		del reloj el número de grados	VIRA 90
		especificado.	Gira la tortuga 90°
			hacia la derecha
GIRAIZQUIERDA	GI	La tortuga gira en sentido	GIRAIZQUIERDA 90 ó
ángu1o		contrario a las agujas	G I 90
		del reloj el ángulo	Gira 1a tortuga 90°
		especificado.	hacia la izquierda
PONLAPIZ	PLA	Pone el lápiz BAJADO y	PONLAPIZ Ó
		el modo PINTA (PINTA	PLA
		es el modo normal de	
		la tortuga para dibujar).	
GOMA	GO	Pone el lápiz BAJADO y	GOMA ó
		el modo a BORRA (la tortuga va	GO
		borrando por donde se	
		mueve).	
SUBELAPIZ	SL	Pone el lápiz LEVANTADO,	SUBELAPIZ Ó
		sin cambiar su modo.	SL
BAJALAPIZ	BL	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 [PONCL [Pone el lápiz del color	El color amarillo es:
]			
]	determinado de acuerdo con:	
]	determinado de acuerdo con: PONCOLORLAPIZ [255 000 000]	PONCOLORLAPIZ [255
]	determinado de acuerdo con: PONCOLORLAPIZ [255 000 000] = Rojo	PONCOLORLAPIZ [255 255 000] ó
]	determinado de acuerdo con: PONCOLORLAPIZ [255 000 000] = Rojo PONCOLORLAPIZ [000 255 000]	PONCOLORLAPIZ [255 255 000] ó PNCL[255 255 000]
]	determinado de acuerdo con: PONCOLORLAPIZ [255 000 000] = Rojo PONCOLORLAPIZ [000 255 000] = Verde PONCOLORLAPIZ [000 255 000]	PONCOLORLAPIZ [255 255 000] ó PNCL[255 255 000]
]	determinado de acuerdo con: PONCOLORLAPIZ [255 000 000] = Rojo PONCOLORLAPIZ [000 255 000] = Verde PONCOLORLAPIZ [000 000 255] = Agul	PONCOLORLAPIZ [255 255 000] ó PNCL[255 255 000]
]	determinado de acuerdo con: PONCOLORLAPIZ [255 000 000] = Rojo PONCOLORLAPIZ [000 255 000] = Verde PONCOLORLAPIZ [000 000 255] = Azul	PONCOLORLAPIZ [255 255 000] 6 PNCL[255 255 000]
OCULTATORTUGA] ОТ	determinado de acuerdo con: PONCOLORLAPIZ [255 000 000] = Rojo PONCOLORLAPIZ [000 255 000] = Verde PONCOLORLAPIZ [000 000 255] = Azul Oculta la tortuga (el	PONCOLORLAPIZ [255 255 000] 6 PNCL[255 255 000]
OCULTATORTUGA] от	determinado de acuerdo con: PONCOLORLAPIZ [255 000 000] = Rojo PONCOLORLAPIZ [000 255 000] = Verde PONCOLORLAPIZ [000 000 255] = Azul Oculta la tortuga (el triángulo) en la pantalla.	PONCOLORLAPIZ [255 255 000] ó PNCL[255 255 000] OT
OCULTATORTUGA MUESTRATORTUGA] ОТ МТ	determinado de acuerdo con: PONCOLORLAPIZ [255 000 000] = Rojo PONCOLORLAPIZ [000 255 000] = Verde PONCOLORLAPIZ [000 000 255] = Azul Oculta la tortuga (el triángulo) en la pantalla. Muestra la tortuga (el	PONCOLORLAPIZ [255 255 000] 6 PNCL[255 255 000] OT MT

HAZTORTUGA tortuga		Coloca la tortuga en la	HAZTORTUGA 0 [0 100
[PosX PosY ángulo		posición que le indicamos,	45 [av 100]]
[instrucciones]]		hincándole la posición y el	Este ejemplo coloca
		ángulo de ubicación, así	la tortuga nº 0 en el
		como las instrucciones	unidad 0 del eje
		que va a realizar	"X" v 100 del eie
		1	"Y" después la rota
			en un ángulo de 45° v
			por último avanza ésta
			100 unidades en este
			sentido
			SelleTuo
BORRAPANTALLA	BP	Borra la pantalla y coloca	BORRAPANTALLA Ó
		a la tortuga en el centro.	BP
LIMPIA		Borra la pantalla, pero	LIMPIA
		mantiene a la tortuga en su	
		posición actual en la	
		pantalla.	
ROTULA [texto]	RO []	Escribe un textoen la	ROTULA [Texto de
		dirección en que se	ejemplo] ó
		encuentra la tortuga.	RO [Texto de ejemplo]
		Este texto semuestra en la	
		pantalla de MSWLogo.	
PONGROSOR Fancho	PONG []	Coloca el ancho del	PONGROSOR [50 50] ó
altol		trazado del lápiz v la	PONG [50 50]
		altura especificados MSWLogo	Aumenta el grosor del
		utiliza solamente el	lániz 50 veces su
		valor de la anchura Por	tamaño
		defecto el valor de la	cumuno
		anchura del lániz es l	
PONPOS [X Y]		Coloca la tortuga en las	PONPOS [100 100]
		coordenadas que se indican,	Coloca la tortuga en
		siendo el primer número las	la unidad 100 del eje
		X y el segundo número	"X" y en la unidad
		las Y.	100 del eje "Y"
PERSPECTIVA		Se añade a la tortuga	PERSPECTIVA
		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.	
PONBALANCEO grados		Rota la tortuga a una nuevo	PONBALANCEO 90
ronbilinitelo grados		nosición absoluta según los	Rota la tortuge 000
		posteron absoruta segun 10s	Rola la colluga 90



		1	
		grados que se indican.	respecto al eje "Y"
		Ésta 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.	respecto al eje "X"
		Ésta 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
		tortuga los grados que	BAJAN 90
		se le indiquen. Este	Desciende 90° el
		comando sólo funciona en modo	sentido de la tortuga
		perspectiva.	

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

PRIMITIVA	MÉTODO ABREVIADO	DESCRIPCIÓN	E IEMPLO

Interfaz de control de dispositivos externos por ordenador a través de puerto paralelo

MUESTRA texto	Escribe la entrada	MUESTRA "ejemplo
	entradas en la salid	ιÓ
	de escritura que este	MUESTRA [Esto es un
	actualmente	ejemplo]
	(inicialmente es el	
	terminal). El texto	
	si es una sola	L I I I I I I I I I I I I I I I I I I I
	palabra llevará	
	comillas de apertur	ı
	", y si es una cadena	ı
	de texto va entre	2
	corchetes[],mostrándo	
	se también los	
	corchetes.	



ESCRIBE texto	ES	Escribe la entrada o	ESCRIBE "ejemplo
		entradas en la salida	ó
		de escritura que	ESCRIBE [Esto es un
		actualmente esté	eiemplol
		inicialmente es el	ó
		terminal). Todas las	ES "eiemplo
		entradas de texto se	ó
		escriben en una sola	ES [Esto es un
		línea Fl texto si	eiemplo]
		es una sola palabra	ejempioj
		llevará comillas de	
		aporturo " y gi	
		apertura , y Si	
		es una cadena de	
		corcnetes[],no	
		mostrandose los	
		corchetes.	
MUESTRAT texto		Muestra la entrada o	MUESTRAT "ejemplo
		entradas similar al	ó
		comando "escribe",	MUESTRAT [Esto es
		excepto que no	un ejemplo]
		muestra el carácter	
		de nueva línea al	
		final, y las entradas	
		múltiples no son	
		separadas por líneas.	
EDITA	EDITAPROCEDIMIENTOS	Abre la ventana del	Este comando se
PROCEDIMIENTOS		editor que nos permite	explica en el tema
		crear o modificar	3 de este manual
		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 manual
		por compatibilidad con	
		versiones anteriores de	
		MSWLogo.	
EDITA "nombre_proc	ED	Abre la ventana del editor con el nombre de procedimiento que se le haya dado.	Este comando se explica en el tema3 de este manual

EDITAFICHERO		Abre un editor estándar	EDITAFICHERO
"nombre_fich		con el fichero que se haya	"c:\\ejemplo.lgo rotula [Este el
11amada		indicado. En llamada se	fichero a editar]
		pone un conjunto de	Abre el fichero "ejemplo.lgo" que
		instrucciones a ejecutar	se encuentra en C, y una vez abierto
		cuando se sale del editor.	muestra en la pantalla de
			MSWLogo el texto del comando
			"rotula"
TAPADO		Devuelve una lista de	Mediante la ayuda del
		contenidos con los	comando "muestra"
		elementos tapados del	nos dice los
		entorno de trabajo.	procedimientos que
			se han ocultado a
			MSWLogo. Con
			respecto al ejemplo
			anterior:
			MUESTRA TAPADO
			Devuelve encender
BORRA contenidos	BO	Borra del entorno de	BORRA [encender]
		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.1go
		para a continuación	Carga el fichero
		realizarlas.	"ejemplo.1go" que
			se encuen tr a en C
GUARDA		Guarda en un fichero	GUARDA
"nombre_fichero		"lgo" las definiciones	"C:\\ejemplo.lgo
		de todos los	Guarda en el
		procedimientos,	fichero
		variables y listas de	"ejemplo.lgo" que
		propiedades sin tapar.	se encuentra en C



BOARCHIVO	BA	Borra el fichero que	BOARCHIVO
"nombre_archivo		se indica, para que funcione el comando el	"C:\\ejemplo.lgoó
		fichero ha de estar	BA"C:\\ejemplo.lgo
		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 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 "I
M1 "D	Hacer girar en sentido 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 "D
М1 "Р	Desactiva el actuador que está conectado a una de las salidas digitales 0 y 1	М1 "Р

M2 "I, M2 "D, M2	Realiza las mismas acciones	M2 "I, M2 "D,
"P M3 "I, M3 "D,	que las primitivas anteriores,	M2 "P M3 "I, M3
M3 "P M4 "I, M4 "D,	lo único que cambia son las	"D, M3 "P M4 "I,
M4 "P	salidas digitales sobre las que	M4 "D, M4 "P
	actúan.	
	$M_2 - Salida 2 \times 3$	
	M3 = Salida 4 y 5	
	MA Salida 6 y 7	
NO	M4 – Saffua 0 y /	
M?	Devuelve una lista con el	Esta primitiva se utiliza
	estado de los 4 motores.	junto con la primitiva
	Los 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 los	M [I D P
	4 motores de manera	P]
	simultáneamente. En la lista de	Se pone entre corchetes
	motores se pone el estado que se	el estado en que queremos
	quiera por cada motor, separados	que encuentren los 4
	éstos por un espacio. La lista ha	motores.
	de ir entre corchetes [].	
CONECTAR	Activa todas las salidas	CONFCTAR
	digitales de la controladora.	Condomix
DESCONECTAR	Desactiva todas las salidas	DESCONECTAR
	digitales de la controladora.	
CONECTA numero	Activa la salida digital indicada	CONECTA 1
CONECTA numero	Activa la salida digital indicada en :NUM(1,2,3,4,5,6,7,8)	CONECTA 1 Activa la salida digital 0
CONECTA numero DESCONECTA numero	Activa la salida digital indicada en :NUM(1,2,3,4,5,6,7,8) Desactiva la salida digital	CONECTA 1 Activa la salida digital 0 DESCONECTA 8
CONECTA numero DESCONECTA numero	Activa la salida digital indicada en :NUM(1,2,3,4,5,6,7,8) Desactiva la salida digital indicada en :NUM	CONECTA 1 Activa la salida digital 0 DESCONECTA 8 Desactiva la salida digital
CONECTA numero DESCONECTA numero	Activa la salida digital indicada en :NUM(1,2,3,4,5,6,7,8) Desactiva la salida digital indicada en :NUM (1,2,3,4,5,6,7,8)	CONECTA 1 Activa la salida digital 0 DESCONECTA 8 Desactiva la salida digital 7
CONECTA numero DESCONECTA numero CONECTADO?	Activa la salida digital indicada en :NUM(1,2,3,4,5,6,7,8) Desactiva la salida digital indicada en :NUM (1,2,3,4,5,6,7,8) Devuelve una lista de 4	CONECTA 1 Activa la salida digital 0 DESCONECTA 8 Desactiva la salida digital 7 Esta primitiva se utiliza
CONECTA numero DESCONECTA numero CONECTADO?	Activa la salida digital indicada en :NUM(1,2,3,4,5,6,7,8) Desactiva la salida digital indicada en :NUM (1,2,3,4,5,6,7,8) Devuelve una lista de 4 elementos (uno por cada	CONECTA 1 Activa la salida digital 0 DESCONECTA 8 Desactiva la salida digital 7 Esta primitiva se utiliza junto con la primitiva
CONECTA numero DESCONECTA numero CONECTADO?	Activa la salida digital indicada en :NUM(1,2,3,4,5,6,7,8)Desactiva la salida digital indicada en :NUM (1,2,3,4,5,6,7,8)Devuelve una lista de 4 elementos (uno por cada motor) cuyos valores son 0 ó 1	CONECTA 1 Activa la salida digital 0 DESCONECTA 8 Desactiva la salida digital 7 Esta primitiva se utiliza junto con la primitiva "muestra", para ver el
CONECTA numero DESCONECTA numero CONECTADO?	Activa la salida digital indicada en :NUM(1,2,3,4,5,6,7,8) Desactiva la salida digital indicada en :NUM (1,2,3,4,5,6,7,8) Devuelve una lista de 4 elementos (uno por cada motor) cuyos valores son 0 ó 1 según estén activos o no.	CONECTA 1 Activa la salida digital 0 DESCONECTA 8 Desactiva la salida digital 7 Esta primitiva se utiliza junto con la primitiva "muestra", para ver el resultado en pantalla.
CONECTA numero DESCONECTA numero CONECTADO?	Activa la salida digital indicada en :NUM(1,2,3,4,5,6,7,8) Desactiva la salida digital indicada en :NUM (1,2,3,4,5,6,7,8) Devuelve una lista de 4 elementos (uno por cada motor) cuyos valores son 0 ó 1 según estén activos o no.	CONECTA 1 Activa la salida digital 0 DESCONECTA 8 Desactiva la salida digital 7 Esta primitiva se utiliza junto con la primitiva "muestra", para ver el resultado en pantalla. MUESTRA CONECTADO?
CONECTA numero DESCONECTA numero CONECTADO?	Activa la salida digital indicada en :NUM(1,2,3,4,5,6,7,8) Desactiva la salida digital indicada en :NUM (1,2,3,4,5,6,7,8) Devuelve una lista de 4 elementos (uno por cada motor) cuyos valores son 0 ó 1 según estén activos o no.	CONECTA 1 Activa la salida digital 0 DESCONECTA 8 Desactiva la salida digital 7 Esta primitiva se utiliza junto con la primitiva "muestra", para ver el resultado en pantalla. MUESTRA CONECTADO? nos devuelve: M [1 0 1 1]
CONECTA numero DESCONECTA numero CONECTADO? SALIDA dato	Activa la salida digital indicada en :NUM(1,2,3,4,5,6,7,8) Desactiva la salida digital indicada en :NUM (1,2,3,4,5,6,7,8) Devuelve una lista de 4 elementos (uno por cada motor) cuyos valores son 0 ó 1 según estén activos o no.	CONECTA 1 Activa la salida digital 0 DESCONECTA 8 Desactiva la salida digital 7 Esta primitiva se utiliza junto con la primitiva "muestra", para ver el resultado en pantalla. MUESTRA CONECTADO? nos devuelve: M [1 0 1 1] SALIDA 131
CONECTA numero DESCONECTA numero CONECTADO? SALIDA dato	Activa la salida digital indicada en :NUM(1,2,3,4,5,6,7,8) Desactiva la salida digital indicada en :NUM (1,2,3,4,5,6,7,8) Devuelve una lista de 4 elementos (uno por cada motor) cuyos valores son 0 ó 1 según estén activos o no.	CONECTA 1 Activa la salida digital 0 DESCONECTA 8 Desactiva la salida digital 7 Esta primitiva se utiliza junto con la primitiva "muestra", para ver el resultado en pantalla. MUESTRA CONECTADO? nos devuelve: M [1 0 1 1] SALIDA 131 Activa las salidas digitales
CONECTA numero DESCONECTA numero CONECTADO? SALIDA dato	Activa la salida digital indicada en :NUM(1,2,3,4,5,6,7,8) Desactiva la salida digital indicada en :NUM (1,2,3,4,5,6,7,8) Devuelve una lista de 4 elementos (uno por cada motor) cuyos valores son 0 ó 1 según estén activos o no.	CONECTA 1 Activa la salida digital 0 DESCONECTA 8 Desactiva la salida digital 7 Esta primitiva se utiliza junto con la primitiva "muestra", para ver el resultado en pantalla. MUESTRA CONECTADO? nos devuelve: M [1 0 1 1] SALIDA 131 Activa las salidas digitales



ENVIAOCTETO dato	Procedimiento para la	ENVIAOCTETO 193
	compatibilidad con las primitivas	Activa las salidas
	ENCONOR (realiza la misma función	digitales
	que la primitiva SALIDA). Envía	0, 6 y 7 (1+64+128)
	a las salidas el número decimal	
	que se ponga en "dato".	
APAGA numero_salida	Procedimiento para la	APAGA 4
	compatibilidad con las primitivas	Desactiva la salida digital
	ENCONOR. Desconecta la salida	número 4
	digital indicada "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
	elementos con el estado de las	MUESTRA SALIDA?
	ocho salidas digitales	nos devuelve:
	(devuelve un 1 si la salida	[1 0 0 0 0 0 1 1]
	está activada o un 0 si la	
	salida se encuentra	
	desactivada.	
VS?	Devuelve en un número decimal	Se utiliza junto con
	el valor en decimal delas	MUESTRA VS?
	salidas digitales que están	nos devuelve: 193
	activadas.	

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 nos
	digital indicada en	devuelve: FALS0
	"numero_entrada" está	Esto ocurriría si la entrada
	conectada y devuelve FALSO	digital 1 no estuviera
	si no está conectada.	

VE?	Devuelve un número	Se utiliza junto con la
	decimal que indica que	primitiva "muestra" para
	entradas digitales están	mostrar el resultado en
	activadas.	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
	digitalque se indica en	primitiva "muestra" para
	"numero_entrada", devuelve	mostrar el resultado en
	un l si la entrada está	pantalla.
	activa y 0 si la entrada no	MUESTRA SD1
	está activa.	0
		Nos indica que la entrada
		digital 0 está desactivada
SD?	Devuelve el estado de	Se utiliza junto con la
	todas las entradas	primitiva "muestra" para
	digitales en forma de lista	mostrar el resultado en
	de Os o 1s.	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	ESPERAON 1
	la espera de que se active	Deja el programa detenido
	la entrada digital indicada	hasta que se active la
	en "numero_entrada".	entrada digital 0
ESPERAOFF numero_entrada	Deja el programa parado a	ESPERAOFF 1
	la espera de que se	Deja el programa detenido
	desactive la entrada	hasta que se desactive la
	digital indicada en	entrada digital 0
	"numero_entrada".	

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

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

SAX	Primitiva valida tanto para	Se utiliza junto conla
	la controladora CNICE como la	primitiva "muestra" para
	ENCONOR.	mostrar el resultado en
	Devuelve el valor en	pantalla.
	voltios de la entrada, esta	MUESTRA SAX
	entrada depende de la	0
	controladora que se esté	Nos indica que la entrada
	utilizando. Será:	analógica 3 de la
	-CNICE: entrada analógica 2	controladora Enconor o la
	-ENCONOR: entrada analógica 3	entrada 2 de la
		controladora CNICE está
		desactivada
SAY	Primitiva valida tanto para	Se utiliza junto conla
	la controladora CNICE como la	primitiva "muestra" para
	ENCONOR -	mostrar el resultado en
	Devuelve el valor en	pantalla.
	voltios de la entrada, esta	MUESTRA SAY
	entrada depende de la	0
	controladora que se esté	Nos indica que la entrada
	utilizando. Será:	analógica 4 de la
	-CNICE: entrada analógica 3	controladora Enconor o la
	-ENCONOR: entrada analógica 4	entrada 3 de la
		controladora CNICE está
		desactivada
SAZ	Primitiva valida tanto para	Se utiliza junto conla
	la controladora CNICE como la	primitiva "muestra" para
	ENCONOR.	mostrar el resultado en
	Devuelve el valor en	pantalla.
	voltios de la entrada, esta	MUESTRA SAZ
	entrada depende de 1a	0
	controladora que se este	Nos indica que la entrada
	utilizando. Será:	analógica 5 de la
	-CNICE: entrada analógica 4	controladora Enconor o la
	-ENCONOR: entrada analógica 5	entrada 4 de la
		controladora CNICE está
		desactivada



SA?	Devuelve una lista con el	Se utiliza junto conla
	valor en voltios de todos los	primitiva "muestra" para
	sensores analógicos.	mostrar el resultado en
		pantalla.
		MUESTRA SA?
		[0 4.6875 0.234375 0]
		Nos muestra el valor en
		Voltios de las 4 entradas analógicas

SALIDAS ANALÓGICAS (Válidas únicamente para la controladora ENCONOR)

PRIMITIVA	DESCRIPCIÓN	EJEMPLO
SALIDAANALOGICA	Escribe los datos necesarios	SALIDAANALOGICA 1 100
numero_salida valor_salida	para poner en la salida	A la salida analógica 1 se
	analógica que se le indica	le asigna el valor de
	en "numero_salida" el valor	salida 100. Éste valor
	255). Para conocer el	corresponde a un nivel de
	valor en tensión en la	tensión que se calcula con
	salida analógica se utiliza	la fórmula anteriormente
	la siguiente fórmula:	descrita.
	Voltaje(V) =	El valor es: 4 ⁻ 20 Voltios
	(valor_salida/23,8)	
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	le asigna un valor de
	en "valor_salida" (varía	tensión de 2 ⁻⁵ voltios
	entre 0, 10,5).	
VOLTAJE?	Devuelve una lista con el	Se utiliza junto conla
	valor en voltios de todas las	primitiva "muestra" para
	salidas 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
1		

Interfaz de control de dispositivos externos por ordenador a través de puerto paralelo

5.4.- Primitivas de sonido

PRIMITIVA	DESCRIPCIÓN	EJEMPLO
TONO [flujosonido]	Emite un tono desonido.	TONO [700 5000]
	Frecuencia — Hertzios	Emite un tono de sonido
	Duración — Milésimas de	agudo durante un período
	segundo El sonido solo se emite por el altavoz del PC.	de tiempo de 5 segundos.
SUENATONO frecuencia	Función similar a la	SUENATONO [1000 5000]
duracion	anterior, que se	Emite un tono de sonido
	mantiene de versiones	más agudo que con el
	anteriores de MSWLogo.	comando anterior durante
		un período de tiempo de 5
		segundos.
ENCIENDESONIDO frecuencia	Emite un sonido cuya	ENCIENDESONIDO 1000
	frecuencia coincide con el	Emite un tono agudo de
	parámetro que se le pasa. El	manera continua. Este
	sonido continuará hasta que	comando no funciona en
	se detenga el comando	sistemas operativos
	"APAGASONIDO".	Windows 2000 y Windows XP.
	Esta primitiva no funcionará	
	en Sistemas Operativos con	
	núcleos NT, ya que utiliza la	
	primitiva "escribepuerto".	
APAGASONIDO	Quitará un sonido que se haya	APAGASONIDO
	puesto con la primitiva	Este comando no funciona
	"ENCIENDESONIDO frecuencia".	en sistemas operativos
	Esta primitiva no funcionará	Windows 2000 y Windows XP.
	en 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	1+8
	se indiquen. Las opciones	Reproduce el fichero
	se indican con un número	"ejemplo.wav" que se
	entero.	encuentra ubicado en C, y
	0-> Síncrono que no	lo reproduce de manera
	devuelve el control hasta que	continuada dejando el
	se haya terminado	control a MSWLogo
	1-> Asíncrono que devuelve	
	el control inmediatamente	
	mientras 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 valorde 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	La variable "entradas"
instrucciones1]	instrucciones si la condición	se ha creado para guardar
[instrucciones2]	expresada se cumple, y	el valorde las entradas
	realiza el otro grupo de	digitales.
	instrucciones sino se cumple	SISINO
	la condición.	(:entradas_digitales=1)
		[M1 "P] [salida 255]
		Se expresa que si la
		entrada digital 0 está
		activada, se desactivan las
		salidas digitales 0 1,
		pero si no está activada, se
		encienden todas las salidas
		digitales.
HAZ.HASTA [instrucciones]	Repite la lista de	La variable "entradas"
[condicion]	instrucciones tantas veces	se ha creado para guardar
	hasta que se cumpla la	el valor de las entradas
	condición. Primero lee el	digitales.
	conjunto de instrucciones,	HAZ.HASTA
	de esta forma se asegura que	[salida 1]
	las instrucciones se ejecutan	SI[:entradas_digitales=5]
	al menos una vez antes de	[salida 0]]
	comprobar la condicion.	
		Lo .primero que hace .es
		chequear las instrucciones
HASIA [condicion]	Repite la lista de	La variable entradas
	hasta que se cumple la	se na creado para guardar
	condición Si la condición	digitales
	expresada se cumple no se	HASTA
	eiecutan las instrucciones	[:bucle=1]
	ni una 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".
HASTA [condicion] [instrucciones]	Repite la lista de instrucciones tanta veces hasta que se cumpla la condición. Si la condición expresada se cumple no se ejecutan las instrucciones ni una sola vez.	La variable "entradas" se ha creado para guardar el valorde las entradas digitales. HASTA [:bucle=1] [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	Repite la lista de	La variable "entradas" se ha
[instrucciones]	instrucciones tantas veces	creado para guardar el valor
[condicion]	como se de la condición	de las entradas digitales.
	expresada. Primero lee el	HAZ_MIENTRAS
	conjunto de instrucciones,	E
	de esta forma se asegura quelas	SI [:entradas_digitales=3]
	instrucciones se ejecutan al	[salida 1]
	menos una vez antes de	SI [:entradas_digitales=5]
	comprobar la condición.	[salida 0]]
		[:buc1e=1]
		Lo primero que hace es
		chequear las instrucciones,
		que son 2 condiciones de los
		"SI", y después comprueba
		sise cumple la condición del
		"HAZ.MIENTRAS", si ésta no
		se cumple no realiza más
		veces las instrucciones del
		"HAZ.MIENTRAS".
MIENTRAS [condicion]	Repite la lista de	La variable "entradas"
	se de la condición. Si la	se ha creado para guardar
	condición expresada no se	el valor de las entradas
	instrucciones ni una sola	digitales.
	vez.	MIENTRAS [:bucle=1] [SI
		[:entradas_digitales=3]
		[salida 1]
		SI [:entradas_digitales=5]
		[salida 0]]
		Primero comprueba sise
		se cumple evalúa las
		instrucciones del "MIENTRAS"
SIEMPRE [instrucciones]	Repite la lista de	SIEMPRE[escribe [Esto
	instrucciones de forma	es un ejemplo]
	repetida.	Esto es un ejemplo
		-
		-
		-
		Escribirá de manera
		continua estas frases,
		hasta que sedetenga el
		proceso mediante la ayuda
		del botón "ALTO".

REPITE numero [instrucciones]	Repite instruc	1a ciones	lista tantas	a de veces	REPITE 3 [escribe [Esto es un ejemplo]]
	como se	indica	a en el	número.	Esto es un ejemplo
					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	Crea una ventana en la	CREAVENTANA "principal"ej
"nombre [titulo] pox poxy	pantalla de MSWLogo.	[Ejemplo de ventana] 120
ancho alto [configuración]	-"padre" es el nombre	150 287 100 []
	principal de la ventana.	Esto creará una ventana
	-"nombre" se utiliza	con el nombre de "ej" en la
	-"pox posy" son las	cual se muestra la
	posiciones de las X y	etiqueta de "Ejemplo de
	-"ancho alto" de la	ventana".
	nueva ventana.	El nombre del proceso
	-"configuración" es una	padre de esta ventana se
	lista de instrucciones, esta	llama "principal", si
	lista puede estar vacía y	esta ventana se pusiera
	añadir los controles más	dentro de otra ventana se
	tarde.	pondría en nombre del
		"padre" de la ventana de la
		cuelga.



BORRAVENTANA "nombre	Borra la ventana que se indica en "nombre".	BORRAVENTANA "principal
		Borra la ventana padre
		de todas las ventanas,
		bo rraría ta mbién las
		ventanas que colgarán de
		esta ventana padre.
		BORRAVENTANA "ej
		Borra solamente la ventana
		etiquetada con el nombre
		"ej".
CREAGROUPBOX "padre	Crea una caja dentro de una	CREAGROUPBOX"ej
"nombre posx posy ancho	ventana en la que se pueden	"ej_groupbox 5 0 274
alto	agrupar botones, botones de	85
	radio, botones de	Crea una caja dentro de la
	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 identificar el "groupbox" y hade ser único.	
	-"pox posy" son las	
	posiciones de las Xylas Y	
	respectivamente de la	
	esquina superior izquierda	
	de izquierda del nuevo	
	"groupbox.	
BORRAGROUPBOX "nombre	Borra el groupbox que se	BORRAGROUPBOX "ej_groupbox
	indica en "nombre".	Borra la caja etiquetada
		como "ej_groupbox".

CREABOTON "padre "nombre	Crea un botón dentro de	CREABOTON "ej "ej_boton
posx posy ancho alto	-"padre" es el nombre de	[Botón] 25 10 110 35
[11amada]	-"nombre" se u tili za para	
	identificar el botón y ha	Se crea un botón dentro
	de ser único.	de la ventanaetiquetada con
	-"pox posy" son las	"ej" alcual se etiqueta
	posiciones de las Xylas Y	con el nombre de
	respectivamente de la	"ej_boton", y muestra
	esquina superior izquierda	el texto de "Botón".
	-"ancho alto" del nuevo	
	botón creado.	
	- "11amada" es una	
	pequeña lista de	
	instrucciones o un nombre	
	de un procedimiento a llamar	
ACTUAL I ZABOTON "nombre	Sustituve el texto que	ACTUALIZABOTON "ei boton
[texto]	aparece dentro del botón.	[Botón
	-"nombre" se utiliza para	actualizado]
	saber que "ventana" cuelga.	Cambia el texto del botón
		etiquetado con "ej_boton"
		por la palabra "Escribe"
		en vez de por la palabra
		"Botón actualizado".
ROPPAROTON "nombro	Porre al hotán que se	ROPPAROTON " ai hotop
BORRADOTON HOMOLE	indica en "nombre".	Borra al batén atiquatada
		come "ai botor"
1		



CREABOTONRADIO "padre "grupo "nombre [etiqueta]	Crea un botón que da al usuario la posibilidad de	CREABOTONRADIO "ej "ej_groupbox "ej_botonradio [Esto es
posx posy ancho alto	selección de un elemento de 2	otro ejemplo] 25 50 120 30
	falso). Este tipo de botón	Crea un botón de
	tiene que estar asociado con	selección dentro del
	un "groupbox".	groupbox al que se
	-"padre" es el nombre de la	etiqueta con
	ventana en la que crea este	"ej_botonradio", el cual
	botón.	tendrá el texto "Esto es un
	-"grupo" es el nombre del	ejemplo".
	"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	
	boton creado.	
BORRABOTONRADIO "nombre	Borra el botón que se	BORRABOTONRADIO
	indica en	"ej_botonradio
	nombre .	Borra el botón etiquetado
		como "ej_botonradio".

CREACHECKBOX "padre	Crea un botón que da al	CREACHECKBOX "ej
"grupo "nombre [etiqueta]	usuario la posibilidad de	"ej_groupbox "ej_checkbox
posx posy ancho alto	selección de un elemento.	[Texto de ejemplo] 155
	Este tipo de botón tiene	10 110 35
	que estar asociado con un	Crea el botón "checkbox"
	"groupbox".	dentro del "groupbox" con
	-"padre" es el nombre de la	nombre de "ej checkbox"
	ventana en la que crea este	en el cual se muestra
	botón.	"Texto de ejemplo".
	-"grupo" es el nombre del	5
	"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 bo t ón.	
	-"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.	
BORRACHECKBOX "nombre	Borra el botón que se	BORREACHECKBOX "ej_checkbox
	"nombre".	Borra el botón etiquetado
		como "ej_checkbox".
		_
CREALISTBOX "padre	Crea una lista de selección	CREALISTBOX "ej "ej_listbox
"nombre posx posy	de diferentes elementos.	155 45 90 20
ancho alto	-"padre" es el nombre de la	Crea una lista de selección
	ventana en la que crea este	dentro de la ventana
	bo t ón.	etiquetada como "ej", a esta
	-"nombre" se utiliza para	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 se indica en "nombre".	BORRALISTBOX "ej_listbox Borra la lista de selección etiquetada como "ej_listbox".
CREACOMBOBOX "padre "nombre posx posy ancho alto	Crea una caja que se utiliza 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".
BORRACOMBOBOX "nombre	Borra la caja de selección de múltiples elementos que se indica en "nombre".	BORRACOMBOBOX "ejcombobox Borra la caja de selección etiquetada como "ej_combobox".
CREAESTATICO "padre "nombre [texto] posx posy ancho alto	Permite mostrar dentro de una ventana un texto. Este texto se puede actualizar con la primitiva "actualizaestatico". -"padre" es el nombre de la ventana en la que se crea. -"nombre" se utiliza para identificar el texto que se va a introducir en la ventana y ha 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.	CREAESTATICO "ej "ej_estatico [ESTATICO] 35 70 50 10 Crea una ventana de texto dentro de la ventana "ej" a la cual se llama "ej_estatico". Esta ventana contiene el texto "ESTATICO".

ACTUALIZAESTATICO "nombre	Sustituye el texto que se	ACTUALIZAESTATICO
[texto]	crea con "creaestatico".	"ej_estatico [ACTUALIZADO]
	-"nombre" se utiliza para	Actualiza el texto de la
	saber de que "creaestatico"	ventana de texto estática de
	cuelga.	"ESTATICO" a ACTUALIZADO".
BORRAESTATICO "nombre	Borra el botón estático	BORRAESTATICO "ej_estatico
	que se 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

jemplo de v entana	

• Crear groupbox

E	Ejemplo de ventana		



• Crear botón

Ejemplo de ven	tana		
	Botón		

Ejemplo) de ventana	
	Botón actualizado	

• Crear botón radio

Ejemplo de ventana	
Escribir	
C Esto es otro ejemplo	

Interfaz de control de dispositivos externos por ordenador a través de puerto paralelo

• Crear checkbox

Escribir Escribir	Texto de ejemplo

• Crear combobox

Ejemplo	de ventana	
	Escribir	🗖 Texto de ejemplo
	O Esto es otro ejemplo	

• Crear estático

Ejemplo de ventana	
Escribir	☐ Te×to de ejemplo
© Esto es otro ejemplo ESTATICO	



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	Devuelve el tamaño de la	Se utiliza junto con la
"ruta_imagen.bmp	imagen que se indica en "ruta_imagen.bmp". El tamaño lo devuelve en una lista de 2 números enteros [Ancho Alto].	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

Alto Trazar Pausa Estado Paso Reiniciar

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.

as=16) [M3 "I] si (:entradas=32) [M3 "D] si (:entradas=5) [M1 "I M2 "D] si (:entradas=9) [M1 "I M2 "I] si
(ve?)
(LEEK_ENIKADA_DIGITAL_BAJA_0) IFER ENTRADA DIGITAL BAJA_outputs_0
(LEER_ENTRADA_DIGITAL_ALTA)
LEER_ENTRADA_DIGITAL_AITA outputs 0
ve? outputs 0
(mientras (:Ducle=0] [ndz entradas ve/si (:entradas=1) [ni 1] si (:entradas=2) [ni D] si (:entradas= mientras outouts [haz "entradas ve/si (:entradas=1) [M1 "I] si (:entradas=2) [M1 "D] si (:entradas=2) [M
as=16) [M3 "1] si (:entradas=32) [M3 "D] si (:entradas=5) [M1 "I M2 "D] si (:entradas=9) [M1 "I M2 "I] si
(ve?)

- 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.

Modo Pausa (Cancelar p	ara Continuar) 🛛 🗙
Entrada:	
OK	Cancel

 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.

Paso simple	X
HAZ "MASCARA	120
Aceptar	Cancelar

- Reiniciar: reinicia la ejecución del programa de MSWLogo que se está ejecutando.
- Ejecutar: e j e c u t a 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".

Guardar co	omo	? ×
Guardar en	n: 🔁 bc5 💽 🖛 🗈 📸 🎫	
Ayuda ENTORM Example Libreria Logolib Traducc	NO es io ion a español	
Nombre:	Guarda	ne
Tipo:	Logo Files (*.LGO)	ar

- 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.

📓 Trabajo		
A	Alto	Trazar
	Pausa	Estado
	Paso	Reiniciar
	1	
guarda "ejemplo.lgd	Ejecutar	Editar

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 "*.*".

Abrir		? ×
<u>B</u> uscar en:	🔁 bc5 💽 🗲 🖻 🕂 🏢]-
Ayuda ENTORI Example Libreria Logolib	NO es io tion a español	
<u>N</u> ombre:	ř.LGO	<u>A</u> brir
Tip <u>o</u> :	Logo Files (*.LGO)	Cancelar

 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.

💯 Trabajo			<u>- </u>
	A	Alto	Trazar
		Pausa	Estado
		Paso	Reiniciar
carga "ejemplo.lgd		Ejecutar	Editar

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 Didáctica del Lenguaje Logo en español:

http://www.quadernsdigitals.net/datos_web/biblioteca/l_8/enLinea/0.htm

• Manual de MSWLogo en español:

http://platea.pntic.mec.es/~lgonzale/tic/programar/manual.html