

## PRIMITIVAS LOGO

### REFERENTES A PROCEDIMIENTOS:

#### <PARA nombre>

Crea un procedimiento o programa llamado nombre.

Las primitivas o/y procedimientos que constituirán las órdenes del que estamos creando las iremos escribiendo en las líneas siguientes. Para terminarlo teclearemos <FIN>

>Para cuadrado :lado

>Repite 4 [av :lado gd 90]

>fin ; para ejecutar este procedimiento, tecleamos su nombre seguido de INTRO

#### <EDITA "nombre"> CON ICONO

Edita el procedimiento llamado "nombre" que habíamos creado previamente. Nos servirá para poder modificarlo.

>edita "cuadrado"

#### <LPROCS> CON ICONO

Lista los procedimientos que tenemos creados en la memoria del ordenador.

>lprocs

#### <ALTO>

Detiene el procedimiento que se está ejecutando y devuelve el control al que lo llamó.

#### <BOTODO> CON ICONO

Borra todos los procedimientos y variables de la memoria, excepto los tapados

## REFERENTES A ENTRADAS Y SALIDAS DEL EXTERIOR:

### <ESCRIBEPUERTO puerto octeto>

Saca por el puerto especificado en "puerto" el valor binario del dato colocado en "octeto". El número de puerto de la salida para la impresora es el 888.

```
>escribepuerto 888 1
```

### <LEEPUERTO puerto>

Devuelve en decimal, el valor binario que hay en el puerto de entrada especificado en "puerto". El puerto de entrada de la impresora es el 889.

```
>leepuerto 889
```

## REFERENTES A OPERACIONES LÓGICAS:

### <O predicado predicado2>

Operación lógica OR.

```
>es o iguales? "A primero "Atención  
devuelve Cierto
```

### <NO predicado>

Es la función lógica NOR.

```
>es no 2>5  
devuelve cierto
```

### <Y predicado1 predicado2>

Es la función lógica Y.

```
>es y "cierto "cierto
```

```
>es y iguales? 5 (8-3) iguales? (10+5) 15  
devuelve cierto
```

## REFERENTES A VARIABLES:

### <HAZ "nombre valor">

Crea la variable llamada "nombre" y le asigna el valor especificado en "valor". Posteriormente nos referiremos a la variable creada anteponiendo : al nombre. Ejemplo:

```
>haz "x 5 ; Asigna 5 a la variable x.  
>Escribe :x ;Nos escribirá el valor de la variable x, o sea 5.
```

## REFERENTES A CONTROL DE FLUJO DE PROGRAMA:

### <REPITE numero [acción]>

Repite el número de veces especificado en "número" la acción que se incluye entre los corchetes. Ejemplo,

>Repite 30 [escribe "hola"] ; Escribe 30 veces la palabra "hola".

### <SI condición [acción]>

Si se cumple la condición especificada en "condición", se ejecuta la acción incluida entre los corchetes, si no se cumple no hace nada.

>si azar 2 = 0 [devuelve "Cero"] [devuelve "Otro"]

### <MIENTRAS predicado [órdenes]>

Ejecuta las órdenes mientras "predicado" sea cierto.

>haz "A 0  
>mientras [:A<200] [av 1 haz "A :A+1"]

### <ESPERA num>

Provoca una detención del programa durante un número de centésimas de segundo especificado en "num". Ejemplo:

>espera 200 ; provoca una detención del programa durante 2 segundos.

## REFERENTES A ENTORNO DE TRABAJO

### <PONCOLORPAPEL num>

Pone el fondo de la *ventana de textos* del color especificado por medio de un número en "num". El número que se puede utilizar está comprendido entre 1 y 255 para una tarjeta VGA, pero puede variar para otras resoluciones. En la versión de Windows se puede utilizar todos los colores que tengamos configurados en la paleta gráfica.

### <PONCOLORTEXTO num>

Ajusta el color del texto de la *ventana de textos* al especificado por medio de un número en "num". El número que se puede utilizar está comprendido entre 1 y 255 para una tarjeta VGA, pero puede variar para otras resoluciones.

### <PONFONDOTEXTO num>

Pone el fondo de los textos de la *ventana de textos* del color especificado por medio de un número en "num". El número que se puede utilizar está comprendido entre 1 y 255 para una tarjeta VGA, pero puede variar para otras resoluciones.

## REFERENTES A ENTRADA Y SALIDA DE DATOS

### <LEECAR>

Queda a la espera de que tecleemos un carácter del teclado. El carácter no se visualiza en pantalla. Ejemplo:  
<Haz "a leecar> Asignará a la variable "a" el carácter que pulsemos en el teclado.

```
>para conducir
>haz "carácter leecar
>si :carácter = "A [avanza 10]
>si :carácter = "B [retrocede 10]
>conduce
>fin
```

### <LEECARS num>

Queda a la espera de que tecleemos el número de caracteres especificado en "num" en el teclado. Los caracteres tecleados no se visualizan en pantalla. Ejemplo:

>Haz "a leecars 8 ; Asignará a la variable "a" la tira de 8 caracteres que pulsemos en el teclado.

### <LEELISTA>

Lee una lista de caracteres introducida por el teclado. Mientras se van tecleando se van visualizando en la pantalla de TEXTOS. Se finaliza la introducción al pulsar la tecla INTRO. Antes de terminar con INTRO nos permite hacer modificaciones en el texto tecleado. Ejemplo:

>Haz "a leelista ; Asignará a la variable "a" la ristra de caracteres que introducimos por el teclado hasta pulsar la tecla INTRO.

### <LEEPALABRA>

Lee una lista de caracteres introducida por el teclado. Mientras se van tecleando se van visualizando en la pantalla de TEXTOS. Se finaliza la introducción al pulsar la tecla INTRO. Antes de terminar con INTRO nos permite hacer modificaciones en el texto tecleado. Solamente captura los caracteres tecleados hasta encontrar un espacio en blanco. Ejemplo:

>Haz "a leelista ; Asignará a la variable "a" la ristra de caracteres que introduzcamos por el teclado hasta pulsar la tecla INTRO. Si la ristra contiene algún carácter "espacio" el resto se ignorará.

### <ESCRIBE "dato">

Sirve para presentar por pantalla el dato presentado en "dato". Si queremos presentar un texto la primitiva será

```
>escribe [este es el texto que quiero presentar]
```

y si lo que queremos es que nos de el valor de una variable

```
>escribe :nombrevariable
```

## REFERENTES A GRÁFICOS:

### <PONFONDO num>

Nos pone el fondo de trabajo de la ventana de gráficos del color especificado en "num". Puede variar entre 1 y 16.

```
>para verfondos
>bp
>bt
>fondos 1
>fin
```

```
>para fondos :color
>si :color = numerocolores + 1 [alto]
>ponfondo :color
>es fondo
>espera 100
>fondos :color+1
>fin
```

### <PONGROSOR num>

Fija el grosor del trazo en la ventana gráfica. "num" puede variar entre 1 (fino) y 10 (muy grueso).

```
>pongrosor 10
```

### <SUBELÁPIZ> SL

Hace que al desplazarnos por la ventana gráfica no se marque el trazo.

### <BAJALÁPIZ> BL

Permite que se vea el trazado al desplazarnos por la ventana gráfica.

### <PONPOS [x y]>

Coloca la tortuga en las coordenadas fijadas en "x" e "y". El punto de referencia es el centro de la pantalla gráfica.

```
>ponpos [60 -80]
```

### <PONCOLORLÁPIZ num>

Determina el color del trazado en la ventana gráfica por medio de "num". Este puede variar entre 1 y 16.

```
>poncolorlapiz 2 ;pone el lápiz de color azul
```

### <RELLENA [x y]>

Rellena del color prefijado la figura cerrada que encierra al punto de coordenadas "x" e "y". Los límites de la figura han de tener el mismo color que el del relleno.

```
>para llenacuadrado
>repite 4 [av 100 gd 90]
>sl
>gd 45
>av 25
>bl
>rellena
>fin
```

### <ROTULA [texto]>

Escribe textos en la *pantalla gráfica*. Hay varios tipos de letra predeterminados. "rotula" utilizará el tipo prefijado previamente.

### <PONTIPO "tipo">

Fija el tipo de letra para la ventana gráfica. Los tipos posibles son (en mayúsculas): CALIGRAFICA DIMINUTA GOTICA ROMAN SANS SANSB TIMES.

### <CARGADECORADO "NOMBRE.GIF">

Carga en la pantalla grafica un decorado.

### VARIAS:

### <TONO frec tiempo>

Produce un sonido en el altavoz del sistema, de la frecuencia indicada en "frec" y del tiempo especificado en "tiempo". Ejemplo

```
>Tono 2000 2 ; Produce un sonido de 2 KHz. durante dos segundos.
```

### <RELOJ>

Da las centésimas de segundo que hace que esta encendido el ordenador.

```
>para tiempo
>haz "tiempo Reloj
>repite 360 [av 1 gd 1]
>es Reloj-tiempo ; tiempo que tarda en ejecutarse la circunferencia
>fin
```