

## INTRODUCCIÓN

MSWLogo es un lenguaje de programación que nos permite construir programas que pueden ser ejecutados por el ordenador. Estos programas están formados por una sucesión de órdenes o instrucciones que el ordenador es capaz de entender y, en consecuencia, realizar las acciones que dichas ordenes le indiquen. Combinando adecuadamente su repertorio de instrucciones podemos hacer juegos, programas matemáticos, gráficos, programas para controlar pequeños motores o bombillas, etc.

En la figura siguiente vemos la pantalla que se obtiene al abrir MSWLogo y una descripción de las zonas en que se divide.



## MOVIENDO LA TORTUGA

Para mover la tortuga y cambiar su dirección Logo cuenta con unas instrucciones muy sencillas que vamos a analizar a continuación. Estas instrucciones y los ejemplos correspondientes podemos probarlos escribiéndolos en la caja de entrada y pulsando ejecutar.

### **borrapantalla**

**bp** Se borra la pantalla de gráficos y la tortuga vuelve a su posición inicial.

### **avanza distancia**

**av distancia** Mueve la tortuga hacia delante la distancia especificada

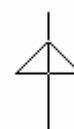
Ejemplo: av 100



### **retrocede distancia**

**re distancia** Mueve la tortuga hacia atrás, en su dirección actual, la distancia especificada.

Ejemplo: av 100 re 50



**giraderecha** ángulo

**gd** ángulo La tortuga gira hacia la derecha (en el sentido de las agujas del reloj) el ángulo especificado, medido en grados.

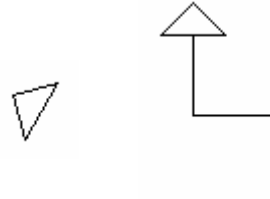
Ejemplo: gd 90  
Ejemplo: av 40 gd 45 av 20



**giraizquierda** ángulo

**gi** ángulo La tortuga gira hacia la izquierda (en el sentido contrario a las agujas del reloj) el ángulo especificado, medido en grados.

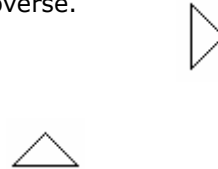
Ejemplo: gi 60  
Ejemplo: av 40 gi 90 av 40 gd 90 av 40



**subelapiz**

**s1** Sube el lápiz haciendo que la tortuga no dibuje al moverse.

Ejemplo: s1 av 50 gd 90 av 50



**bajalapiz**

**b1** Baja el lápiz haciendo que la tortuga dibuje al moverse. Es la situación por defecto al iniciar el programa.

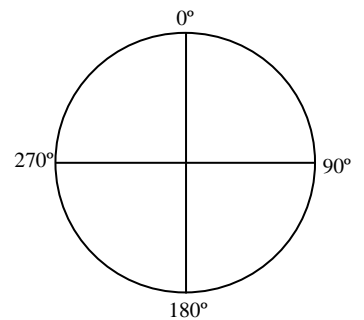
Ejemplo: gd 90 av 50 s1 av 50 b1 av 50



**ponrumbo** ángulo

**ponr** ángulo Cambia el rumbo de la tortuga por un nuevo rumbo absoluto. El argumento es un ángulo, el rumbo se mide en grados según el movimiento de las agujas del reloj desde el eje y.

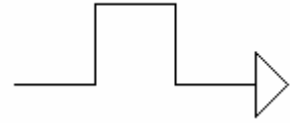
Ejemplo: ponrumbo 90



**ACTIVIDADES:**

1. Comprobar los ejemplos de las instrucciones de movimiento de la tortuga.
2. Escribir las instrucciones necesarias (utilizar av y gd) para dibujar un rectángulo de dimensiones 50 de alto y 100 de largo.

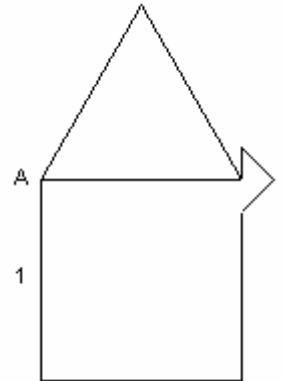
- Introducir las instrucciones necesarias (utilizar av, gi y gd) para realizar el dibujo adjunto. Las líneas son de 40 puntos.



- Completar las instrucciones para dibujar un triángulo equilátero de lado 100.

```
bp gd 30 av 100
gd 120 av
gd av 100
```

- Introducir las instrucciones necesarias para realizar el dibujo adjunto. Las dimensiones de todas las líneas es 100. Comenzar desde el punto A, para dibujar en primer lugar la línea 1.



## PROCEDIMIENTOS

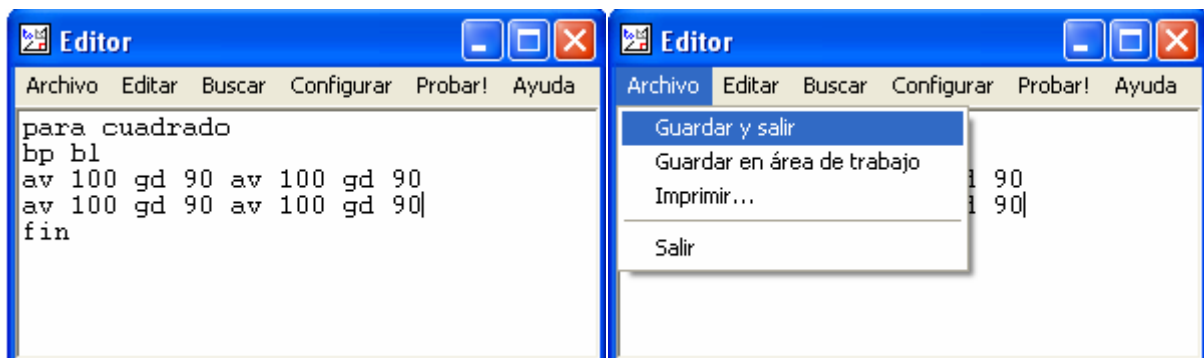
Tal como hemos visto en el apartado anterior, a través de la caja de entrada Logo ejecuta las instrucciones sueltas conforme se van introduciendo. Para crear un verdadero programa tenemos que darle un nombre y conseguir que todas las instrucciones que lo forman se ejecuten cuando escribamos ese nombre en la caja de entrada.

Pues bien, un procedimiento es un conjunto de instrucciones, agrupadas bajo un nombre, que serán ejecutadas al escribir el nombre del procedimiento en la caja de entrada. Su estructura es la siguiente:

```

para nombre del procedimiento
    .....
    instrucciones que lo forman
    .....
fin
    
```

Para crear un procedimiento se utiliza el editor con que cuenta MSWLogo, a él se accede pinchando en el botón Editar, situado en la parte inferior derecha. Como ejemplo vamos a realizar un procedimiento para dibujar un cuadrado de lado 100. Una vez escrito, pulsamos Guardar y Salir para que el procedimiento quede almacenado en la memoria del ordenador y podamos ejecutarlo poniendo su nombre en la caja de entrada.

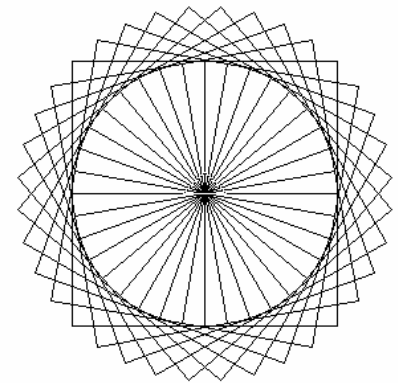




10. Realizar un procedimiento, denominado dibujo, que realice la figura adjunta. Utilizar como base un cuadrado de lado 100 y repetirlo 36 veces girándolo 10° entre cada repetición.

```

para cuadrado           para dibujo
                        repite 36 [
                                ]
fin
                        fin
    
```

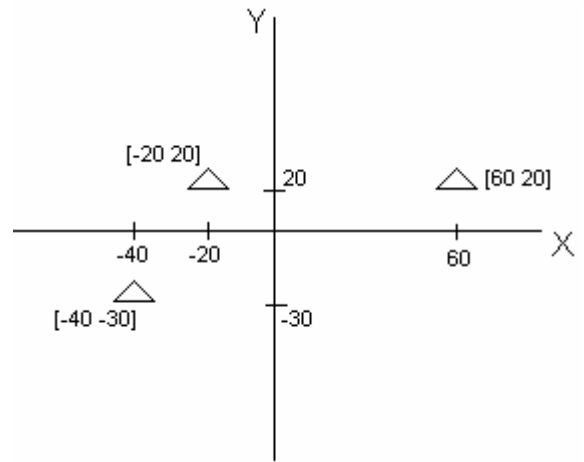


### LAS COORDENADAS DE LA TORTUGA. PONPOS.

En la pantalla podemos imaginar dos ejes de coordenadas estando la tortuga inicialmente en el origen, es decir, en la posición x=0 y=0, en la sintaxis de Logo [0 0].

En la figura podemos ver diversas posiciones de la tortuga y sus coordenadas.

La instrucción que nos permite mover la tortuga utilizando sus coordenadas es:



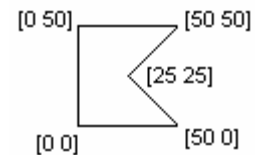
#### **ponpos** [coordenadaX coordenadaY]

Ubica la tortuga en el punto cuyas coordenadas se indican. Si el lápiz está bajado dibuja una línea recta en su desplazamiento.

Ejemplo: Vamos a realizar el dibujo de la figura moviendo la tortuga a través de sus coordenadas:

```

para coordenadas
ponpos [0 50] ponpos [50 50]
ponpos [25 25] ponpos [50 0]
ponpos [0 0]
fin
    
```



### LOS COLORES Y EL GROSOR DEL LÁPIZ.

Veamos por tanto algunas instrucciones para trabajar con colores en Logo:

#### **poncolorlapiz** [rojo verde azul]

poncl [rojo verde azul] Selecciona el color del lápiz. Éste se obtiene mediante una mezcla de los colores rojo, verde y azul, indicando su proporción mediante un número comprendido entre 0 y 255.

negro:	poncl [0 0 0]	violeta:	poncl [255 0 255]
azul:	poncl [0 0 255]	amarillo:	poncl [255 255 0]
verde:	poncl [0 255 0]	blanco:	poncl [255 255 255]
azul claro:	poncl [0 255 255]	gris:	poncl [128 128 128]
rojo:	poncl [255 0 0]		

#### **píxel**

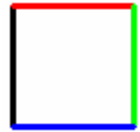
Esta instrucción nos indica el color del punto de la pantalla (píxel) en el que se encuentra la tortuga, mostrándonos los números que representan la intensidad de los colores rojo, verde y azul correspondientes.

**pongrosor** [altura anchura]

pong [altura anchura] Define el grosor del lápiz

Ejemplo: Las siguientes instrucciones dibujan un cuadrado en el que cada lado es de un color, siendo el trazo de grosor 3:

```
para colores
pong [3 3]
poncl [0 0 0] av 60 gd 90
poncl [255 0 0] av 60 gd 90
poncl [0 255 0] av 60 gd 90
poncl [0 0 255] av 50 gd 90
fin
```



También tenemos instrucciones para rellenar de un color determinado una figura cerrada. Se trata de las instrucciones poncolorrelleno y rellena:

**poncolorrelleno** [rojo verde azul]

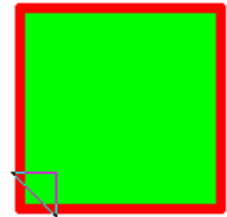
pccr [rojo verde azul] Pone el color de relleno indicado

**rellena**

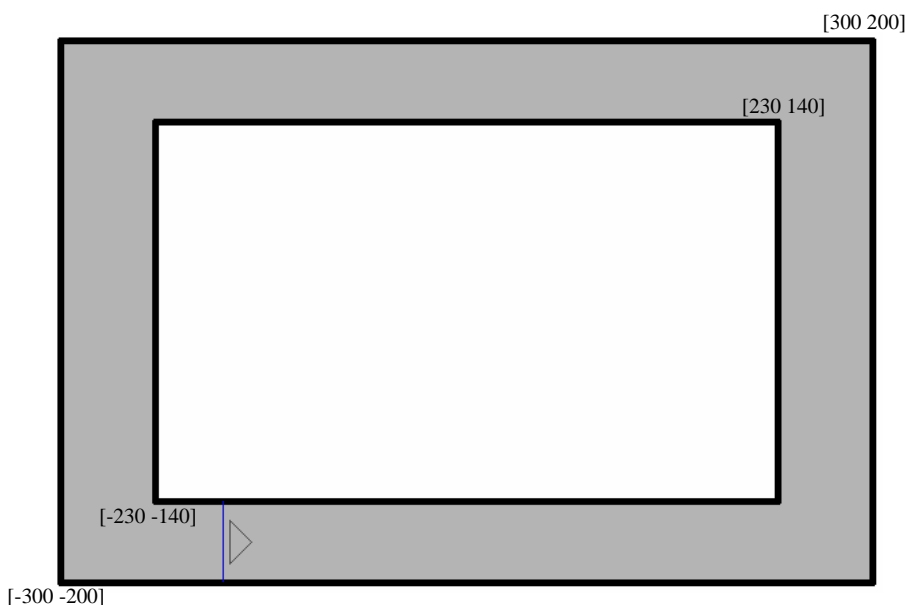
Rellena una parte de la ventana que contiene los gráficos de la tortuga y que se ha limitado con líneas cerradas que se han dibujado previamente. La tortuga debe estar situada en el interior de la figura.

Ejemplo: Vamos a dibujar un cuadrado de color rojo relleno de verde:

```
para cuadradodecolor
poncolorlapiz [255 0 0] pongrosor [5 5]
repite 4[av 100 gd 90]
sl gd 45 av 10 bl
poncolorrelleno [0 255 0] rellena
fin
```

**ACTIVIDADES:**

11. Comprobar los ejemplos de los apartados anteriores.
12. Realizar un procedimiento denominado circuito que realice el dibujo adjunto.



13. El procedimiento adjunto va a provocar que la tortuga esté moviéndose rápidamente, en línea recta, por la pantalla. Queremos conseguir que la velocidad del movimiento sea más lenta, para ello podemos tener en cuenta que el ordenador tarda un cierto tiempo en ejecutar las instrucciones, por tanto, si repetimos “nada” un determinado número de veces conseguiremos provocar un retraso que, en este caso, dependerá de la rapidez del ordenador y habrá que determinar haciendo varias pruebas. Comprobar los procedimientos y hacer las pruebas necesarias para que la tortuga se mueva lentamente.

para movimiento	para movimiento
av 1	av 1
movimiento	repite [ ]
fin	movimiento
	fin

## CONTROL DESDE EL TECLADO

Para poder utilizar el teclado como elemento de control tenemos que usar dos instrucciones ponfoco y ponteclado:

**ponfoco** [nombre de la pantalla que queremos tener activa]

Pasa el control a la pantalla especificada: Pantalla de MSWLogo o Trabajo.

Ejemplo: ponfoco [Pantalla de MSWLogo]

Esta instrucción es necesaria como paso previo a activar los eventos del teclado.

**ponteclado** [instrucciones]

Activa los eventos del teclado, es decir, cada vez que pulsemos una tecla Logo detendrá lo que esté haciendo en ese momento y ejecutará las instrucciones y/o procedimientos escritos en el corchete, posteriormente continuará con el trabajo que había parado.

Ejemplo: ponteclado [gd 20] cada vez que pulsemos una tecla la tortuga gira 20° a la derecha

### ACTIVIDADES:

14. Comprobar el procedimiento denominado juego que contiene las instrucciones ponfoco y ponteclado, de manera que la tortuga girará 20° a la derecha cada vez que pulsemos una tecla. Este mismo procedimiento llama al procedimiento del ejercicio 13 para que la tortuga esté en movimiento.

para juego	para movimiento
ponfoco [Pantalla de MSWLogo]	av 1 repite 40000[ ]
ponteclado [gd 20]	movimiento
movimiento	fin
fin	

15. Escribir un procedimiento de nombre líneas que dibuje una línea vertical de 20 puntos cada vez que pulsemos una tecla (las líneas estarán separadas unas de otras 10 puntos).



## OBTENER Y GUARDAR LA TECLA PULSADA. VARIABLES.

Vamos a ver, a continuación, dos instrucciones que nos permiten conocer la tecla que hemos pulsado y la forma de guardar el carácter correspondiente a la tecla en la memoria del ordenador, dentro de una variable.

**leercar**

lc Indica el código ASCII (número entre 1 y 127) de la última tecla pulsada.



18. Queremos utilizar el dibujo realizado en el ejercicio 12 como un circuito para que la tortuga lo recorra, controlando los giros con las teclas “4” y “6”. Además el programa tiene que controlar que cuando la tortuga choque con un borde negro tiene que volver al punto de partida, para ello cada vez que la tortuga se mueva hay que preguntar por el color del píxel en el que se encuentra.

19. Sabemos que la velocidad de la tortuga depende del número que pongamos en la instrucción repite, cuanto más grande sea más lenta irá la tortuga, ya que el retraso que se produce es mayor. Queremos ampliar el programa anterior para poder **controlar la velocidad de la tortuga**: si pulsamos la tecla x la tortuga se debe mover más rápida, si pulsamos la tecla z el movimiento será más lento. Para conseguirlo, tenemos que crear una variable, denominada por ejemplo “velocidad, cuyo valor cambie al pulsar las teclas x y z; el contenido de esta variable lo pondremos en la instrucción repite para provocar el retraso.

Es decir, con la instrucción haz creamos la variable “velocidad y le damos un valor inicial de 20000. Si pulsamos la tecla x utilizamos la instrucción haz para cambiar el valor de “velocidad a 800 (el movimiento será más rápido) y si pulsamos la tecla z lo cambiamos por 40000 (el movimiento será más lento). En la instrucción repite pondremos el contenido de la variable, poniendo dos puntos delante de su nombre :velocidad.

## DIBUJAR EL CIRCUITO CON PAINT. CARGADIB.

El circuito por el que se va a mover la tortuga también lo podemos dibujar con el programa Paint, teniendo presente que los bordes deben realizarse en color negro y la línea de salida en color azul (del menor grosor posible). Una vez hecho, lo guardamos en la misma carpeta en la que hemos almacenado los procedimientos anteriores con el nombre de "circuito" y la extensión bmp.

Para llevar el dibujo a la pantalla de MSWLogo utilizamos la instrucción cargadib:

**cargadib** nombrebitemap

Carga el bitmap que se indica en la entrada que debe ser una palabra. Esta instrucción es similar a la opción Cargar del Menú Bitmap. Nombrebitemap es el nombre del fichero que contiene el bitmap.

Ejemplo:           cargadib "circuito.bmp           El archivo debe estar en la misma carpeta que el procedimiento que lo utilice.

### ACTIVIDADES:

20. Dibujar un circuito, con el programa Paint, y cargarlo en nuestro juego con la instrucción anterior. Tener en cuenta que habrá que utilizar ponpos y ponrumbo para que, una vez cargado el circuito, coloquemos la tortuga delante de la línea de salida.