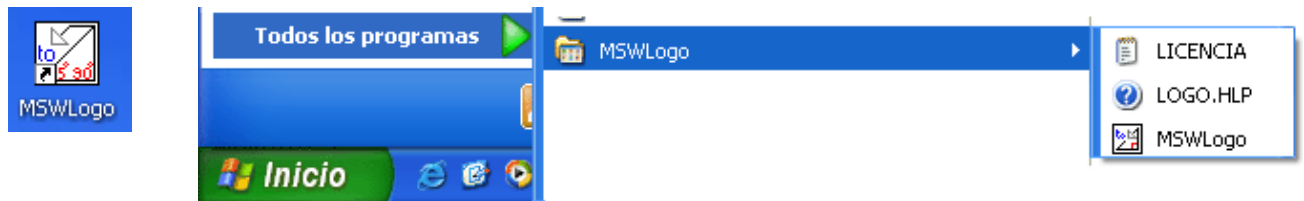


Arrancar el programa.

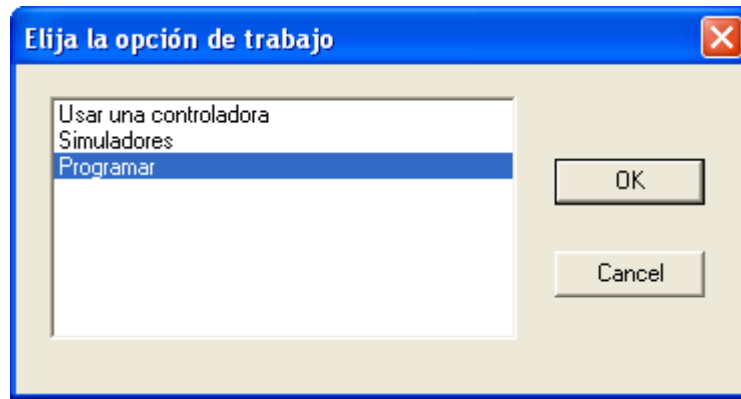
Localiza en el escritorio o en el menú Inicio el icono del programa para poder arrancar el programa



Nos aparece una ventana con detalles acerca de MSWLogo. Pulsamos en OK



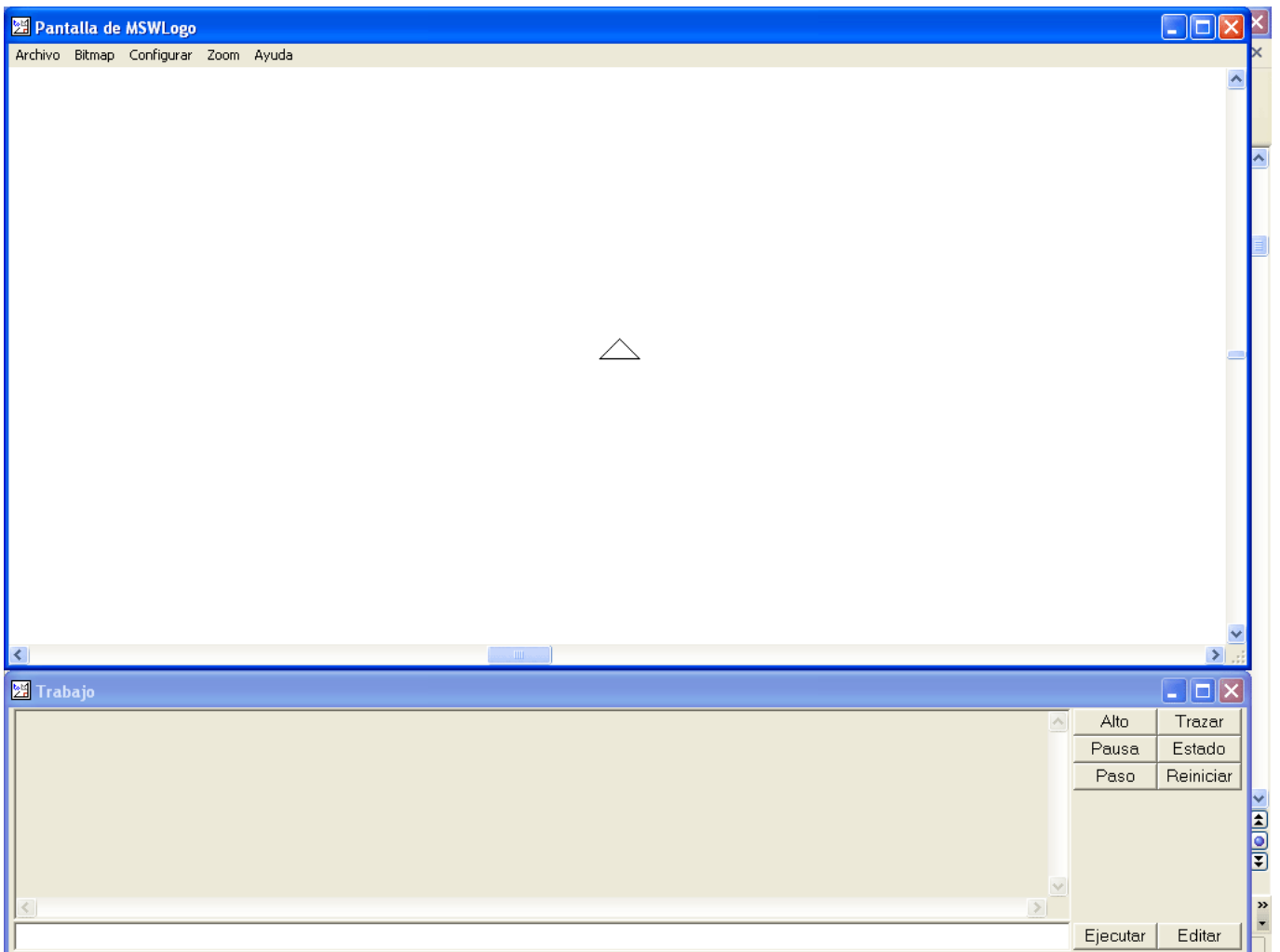
A continuación nos pide que elijamos una opción de trabajo. Elegimos programar y hacemos clic en OK



nos aparece el entorno de trabajo.

El entorno de MSWLogo

Tenemos dos ventanas, la ventana de MSWLogo y la de trabajo.



Pantalla de MSWLogo: en el centro de la misma aparece un triángulo que se desplazará dejando un rastro en función de las primitivas que hayamos escrito en la ventana de trabajo.

Ventana de trabajo: en ella escribiremos las primitivas que realizarán diferentes operaciones. En la parte de abajo existe un cuadro de inserción de texto donde escribiremos las ordenes, tras esto pulsaremos Intro, se ejecutará la orden y ésta pasará a la parte de arriba.

Para salir podemos hacer:

1. escribir la primitiva "ADIOS"
2. Menú Archivo>Salir.
3. Clic en el botón Cerrar de la barra de título.

Movimiento. Primeras primitivas.

Las cosas que sabe hacer MSWLogo se llaman **primitivas** o **procedimientos primitivos** (después veremos otros procedimientos que no vienen de origen con el programa y son la forma de enseñarle más cosas). Para ejecutarlas deberemos escribirlas en la línea de inserción de texto de la ventana de trabajo y al pulsar la tecla Intro el programa realizará la acción asociada a dicha primitiva.

Veamos un ejemplo, escribe en la ventana de trabajo

```
AVANZA 60
GIRADERECHA 90
AVANZA 60
GIRADERECHA 90
AVANZA 60
GIRADERECHA 90
AVANZA 60
GIRADERECHA 90
```

Habrás observado como la primitiva AVANZA hace avanzar el triángulo 60 unidades al tiempo que va dibujando una línea, mientras que la primitiva GIRADERECHA le hace girar sobre si mismo y a la derecha tantos grados como indica el número que lleva a continuación.

Busca en la ayuda información sobre estas dos primitivas para irte familiarizando con el modo en que se describen éstas en el programa (Usando el lenguaje>usando gráficos>movimiento de la tortuga, es un buen sitio para mirar).

Las dos primitivas anteriores pueden escribirse también como AV (AVANZA) y GD (GIRADERECHA).

Escribe ahora otra primitiva en la ventana de trabajo

```
BP
```

Como has podido comprobar se borra cualquier dibujo que haya sobre la ventana de gráficos y la tortuga vuelve a su posición inicial. La primitiva BP es la forma abreviada de BORRAPANTALLA.

Busca información en la ayuda sobre las primitivas RETROCEDE (RE) Y GIRAIZQUIERDA (GI) e intenta dibujar una silla, una F, un ángulo recto, polígonos regulares de 3, 5, 6, 7, 8, 9 y 10 lados (el ángulo que habrá de girar la tortuga para dibujar un polígono de n lados es igual a $360/n$). Procura que la tortuga siempre quede en la posición y en la dirección que tenía al principio.

SILLA	LETRA F	ÁNGULO RECTO	TRIÁNGULO	PENTÁGONO
HEXÁGONO	HEPTÁGONO	OCTÓGONO	ENEÁGONO	DECÁGONO

Iteración.

Al escribir las primitivas para dibujar los polígonos regulares habrás comprobado que hay secuencias que se repiten. Para estos casos MSWLogo me ofrece la primitiva REPITE. Busca información en la ayuda para conocer como funciona.

Intenta dibujar con ayuda de la primitiva repite un cuadrado. Si no lo consigues prueba lo siguiente

```
REPITE 4 [AV 60 GD 90]
```

Intenta dibujar con ayuda de la primitiva REPITE el resto de polígonos regulares.

TRIÁNGULO	PENTAGONO	HEXAGONO
HEPTAGONO	OCTOGONO	ENEAGONO
DECAGONO		

Escribe las ordenes necesarias para dibujar un polígono regular de 360 lados avanzando una unidad para cada lado, ten en cuenta lo dicho anteriormente respecto del ángulo que ha de girar la tortuga para dibujar un polígono de n lados. ¿Se parece a una circunferencia?.

Escribe lo siguiente:

REPITE 90 [AV 1 GD 1]

¿Se parece a un arco de circunferencia? ¿Cuántos grados abarca?.

Estados del lápiz.

Busca información sobre las primitivas SUBELAPIZ (SL) Y BAJALAPIZ (BL) e intenta dibujar dos líneas paralelas, tres triángulos equiláteros en la misma pantalla, un triángulo equilátero un cuadrado y una circunferencia en la misma pantalla, una circunferencia con una recta tangente y otra secante.

DOS LINEAS PARALELAS	TRIANGULO CUADRADO CIRCUNFER.	CIRCUNFERENCIA TG. SECANTE
----------------------	-------------------------------	----------------------------

Color.

Busca información sobre las primitivas COLORLAPIZ (CL) Y PONCOLORLAPIZ (PONCL) y escribe las ordenes necesarias para dibujar un triángulo equilátero de forma que cada uno de sus lados tenga un color diferente.

Dibuja un triángulo y sus tres mediatrices, éstas últimas en distinto color que el triángulo.

Busca información sobre la primitiva RELLENA y escribe las ordenes para dibujar un triángulo un cuadrado y un pentágono rellenos de un color diferente cada uno.

Escribe las ordenes necesarias para dibujar la bandera de Castilla la Mancha, haz lo mismo con la bandera de España.

Castilla Mancha	España
-----------------	--------

A la forma de trabajar hasta aquí se le suele llamar ejecutar ordenes en MODO DIRECTO.

Procedimientos fijos.

Hasta ahora ejecutábamos “procedimientos primitivos” o “primitivas” (acciones que ya sabe hacer MSWLogo). Ahora le vamos a enseñar a hacer cosas que no sabe mediante “procedimientos”.

Definición de Procedimientos

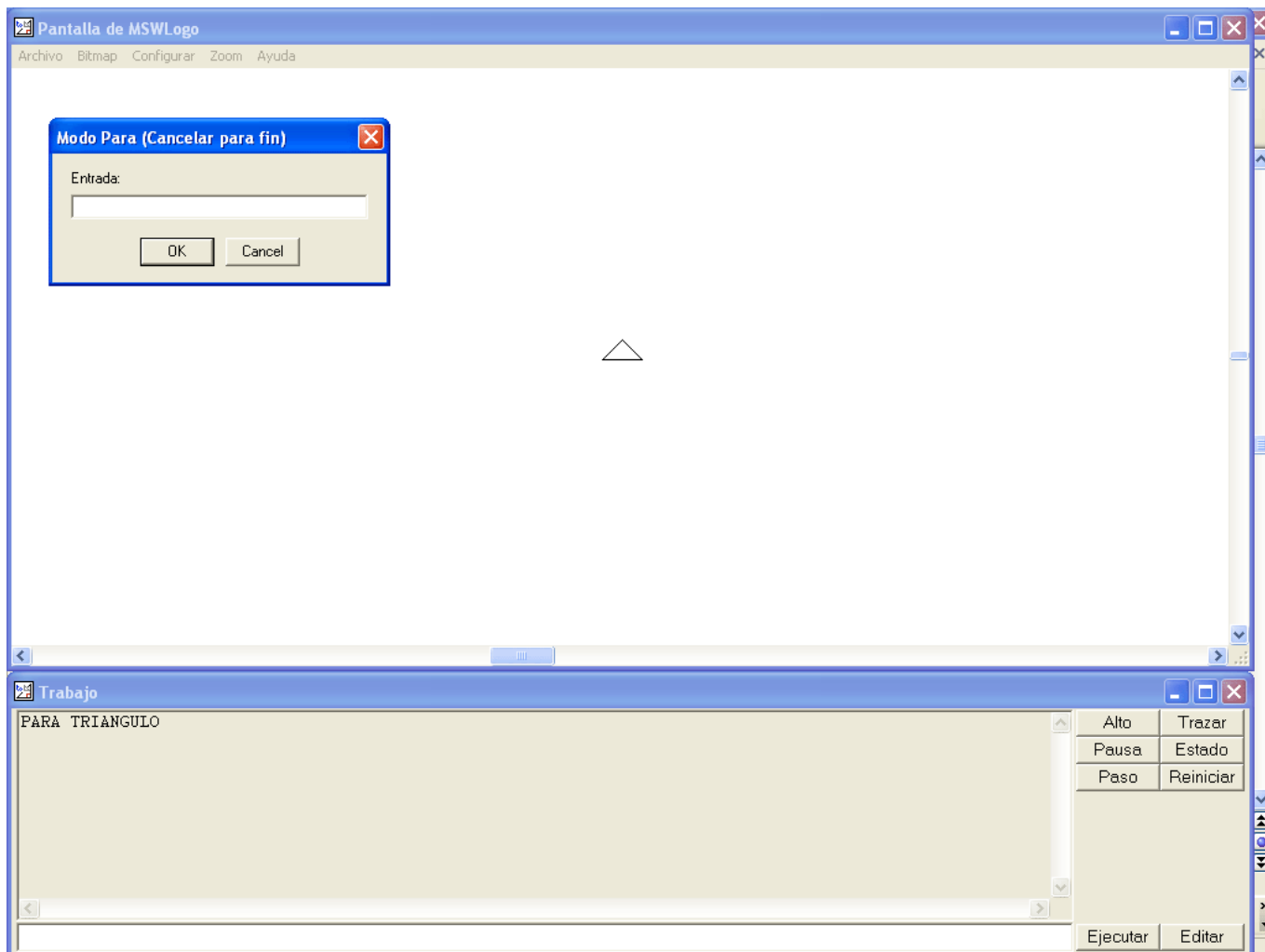
Nuestros procedimientos comenzarán con la primitiva PARA y separado de esta mediante un espacio escribiremos el nombre del procedimiento. Este nombre no podrá coincidir con el de una primitiva ni contener símbolos propios del lenguaje (+, =, etc). En la siguiente línea comenzaremos a escribir las diferentes ordenes que queremos que ejecute el procedimiento y terminaremos el procedimiento con la primitiva FIN. Un ejemplo de procedimiento para dibujar un triángulo podría ser el siguiente:

```
PARA TRIANGULO  
REPITE 3 [AVANZA 40 GIRADERECHA 120]  
FIN
```

Para definir un procedimiento tenemos 2 posibilidades:

Utilizando la ventana de trabajo.

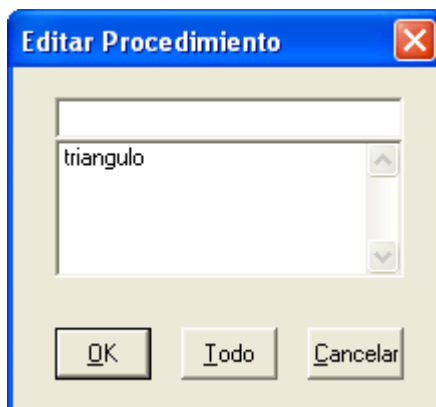
Escribiremos en el cuadro de entrada de texto “PARA (nombre de procedimiento)” y al pulsar ENTER o el botón Ejecutar nos aparecerá la ventana “Modo Para”.



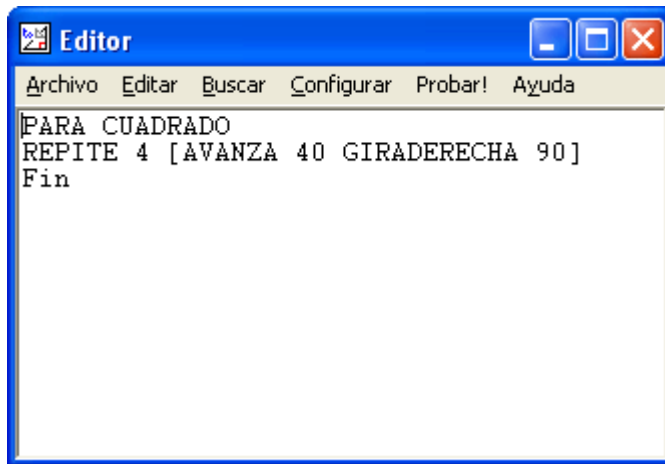
Ahora, en el cuadro de texto Entrada escribiría las instrucciones (REPITE 3 [AVANZA 40 GIRADERECHA 120]) pulsando OK tras cada línea. Una vez introducidas todas las líneas pulsaría Cancel para terminar. En la ventana trabajo me indicará que se ha definido el procedimiento que acabamos de introducir (TRIANGULO definido). Si escribimos “TRIANGULO” en el cuadro de texto de la ventana Trabajo se ejecutarán las ordenes del procedimiento y dibujará un triángulo.

Utilizando la ventana de edición.

Cuando el número de líneas es grande, deseo hacer correcciones en un procedimiento o simplemente quiero copiar y pegar es más flexible utilizar la ventana de edición. En el menú Archivo haz clic en editar. Aparecerá la ventana “Editar Procedimiento”.



Para ver la flexibilidad de este editor haz clic en triángulo y pulsa OK. Te aparecerá la ventana del editor. Modifica el nombre del procedimiento, el número de repeticiones y el ángulo a girar para que ahora dibuje un cuadrado. Tras esto, en el menú Archivo del Editor pulsa en Guardar y salir.



Comprueba que ahora el procedimiento CUADRADO dibuja un cuadrado.

Busca más información sobre estas 2 primitivas (PARA y FIN) en la ayuda de MSWLogo.

Define procedimientos que dibujen los diferentes polígonos regulares de 4 a 10 lados y la circunferencia.

Cuadrado	Pentágono	Hexágono
Heptágono	Octógono	Eneágono
Decágono	Circunferencia	

Busca información sobre los polígonos estrellados e intenta definir procedimientos que dibujen polígonos estrellados.

Guardando y recuperando el trabajo.

Aunque tras la definición de cada procedimiento en la ventana de edición has tenido que pulsar en Guardar y salir, habrás observado que no te ha pedido un nombre de archivo ni un lugar donde guardarlo. Esto es por que MSWLogo te permite guardar varios procedimientos en un mismo archivo, y normalmente los proyectos tienen más de un procedimiento. Imagina que quieres guardar todos los procedimientos que acabas de realizar con el nombre "POLIGONOS". Tras haber creado los procedimientos por el método que desees, en el menú de la ventana "Pantalla de MSWLogo" selecciona Archivo>Guardar, irás a la ventana "Guardar" que ya conoces. Se creará un archivo con el nombre que decidas y la extensión "LGO. A partir de ese momento podrás recuperar todos esos procedimientos cuando desees yendo a menú Archivo>Cargar y elegir el nombre del archivo donde los guardaste.

Guarda los procedimientos CUADRADO, PENTAGONO,DECAGONO Y CIRCUNFERENCIA en un archivo de nombre POLIGONOS.

Procedimientos variables.

De 1 entrada.

Hasta ahora el procedimiento definía el tamaño de las figuras que venimos haciendo. Si quería modificar el tamaño debía editar el procedimiento y realizar las modificaciones pertinentes.

Lo que vamos a hacer ahora es crear procedimientos que llevan implícitos, en la línea de llamada a los mismos, ciertos valores para utilizarlos en la ejecución del mismo.

Veamos un ejemplo

```

PARA HEXAGONO :LADO
REPITE 6 [AV :LADO GD 60]
FIN
    
```

Cuando escriba
 HEXAGONO 60
 HEXAGONO 70
 HEXAGONO 80

MSWLogo dibujará hexágonos de lado 60, 70 y 80 respectivamente.

Escribe procedimientos con una variable para el lado del resto de los polígonos regulares.

Ejecuta el siguiente procedimiento

```
PARA CIRCUNFERENCIA :RADIO
REPITE 360 [AV 2*PI*:RADIO/360 GD 360/360]
FIN
```

¿Qué hace?

Escribe el procedimiento de nombre POLIGONO_REGULAR :LADOS que me dibuje cualquier polígono regular de 40 uds de lado dándole el número de lados que ha de tener.

De varias entradas.

Vista la flexibilidad que me ofrece podríamos pensar en introducirle junto con el nombre del procedimiento más de un valor. La forma de hacerlo es similar.

Prueba lo siguiente

```
PARA RECTANGULO :LADO1 :LADO2
REPITE 2 [AV :LADO1 GD 90 AV :LADO2 GD 90]
FIN
```

Crea un procedimiento de nombre POLIGONO :LADO :ANGULO que dibuje polígonos dado el lado y el ángulo de giro.

Crea un procedimiento de nombre POLIGONO :LADO :LADOS que dibuje polígonos en función del lado y del número de lados.

Prueba lo siguiente

```
PARA ROMBO :LADO1 :LADO2 :ANGULO
REPITE 2 [AV :LADO1 GD :ANGULO AV :LADO2 GD 180-:ANGULO]
FIN
```

Creación de variables y operaciones con variables.

Busca información sobre la primitiva HAZ.

Prueba lo siguiente:

```
PARA ESPIRALCUADRADA :LADO
REPITE 60 [AV :LADO GD 90 HAZ "LADO :LADO+1]
FIN
ESPIRALCUADRADA 5
```

Como habrás observado, el valor introducido para la variable :LADO es un valor de inicio que vamos incrementando a lo largo de la ejecución del procedimiento.

Escribe el procedimiento de nombre ESPIRALCIRCULAR :LADO que dibuje espirales circulares.

Procedimientos que llaman a otros procedimientos.

Lo bueno de los procedimientos no es solo que nos permiten ejecutarlos escribiendo solamente el nombre, además puedo llamarlos desde otros procedimientos haciendo más sencilla y rápida la programación. Veamos un ejemplo

```
PARA CUADRADO :LADO
REPITE 4 [AV :LADO GD 90]
FIN
```

```
PARA DIBUJO
REPITE 6 [AV 20 GD 60 CUADRADO 75]
FIN
```

Como habrás observado, el procedimiento dibujo llama al procedimiento cuadrado.

Haz pruebas variando el polígono dibujado, el número de veces que se repite y la posición que toma antes de dibujar el polígono.

Recursión.

Hasta ahora controlábamos el número de ciclos que se ejecutaba una serie de ordenes mediante la primitiva REPITE, pero a veces necesitamos que el programa esté ejecutándose de forma permanente. En estas situaciones podemos hacer que el procedimiento se llame a si mismo. A esto se le llama **recursión**.

Veámoslo con un ejemplo que ya debes conocer

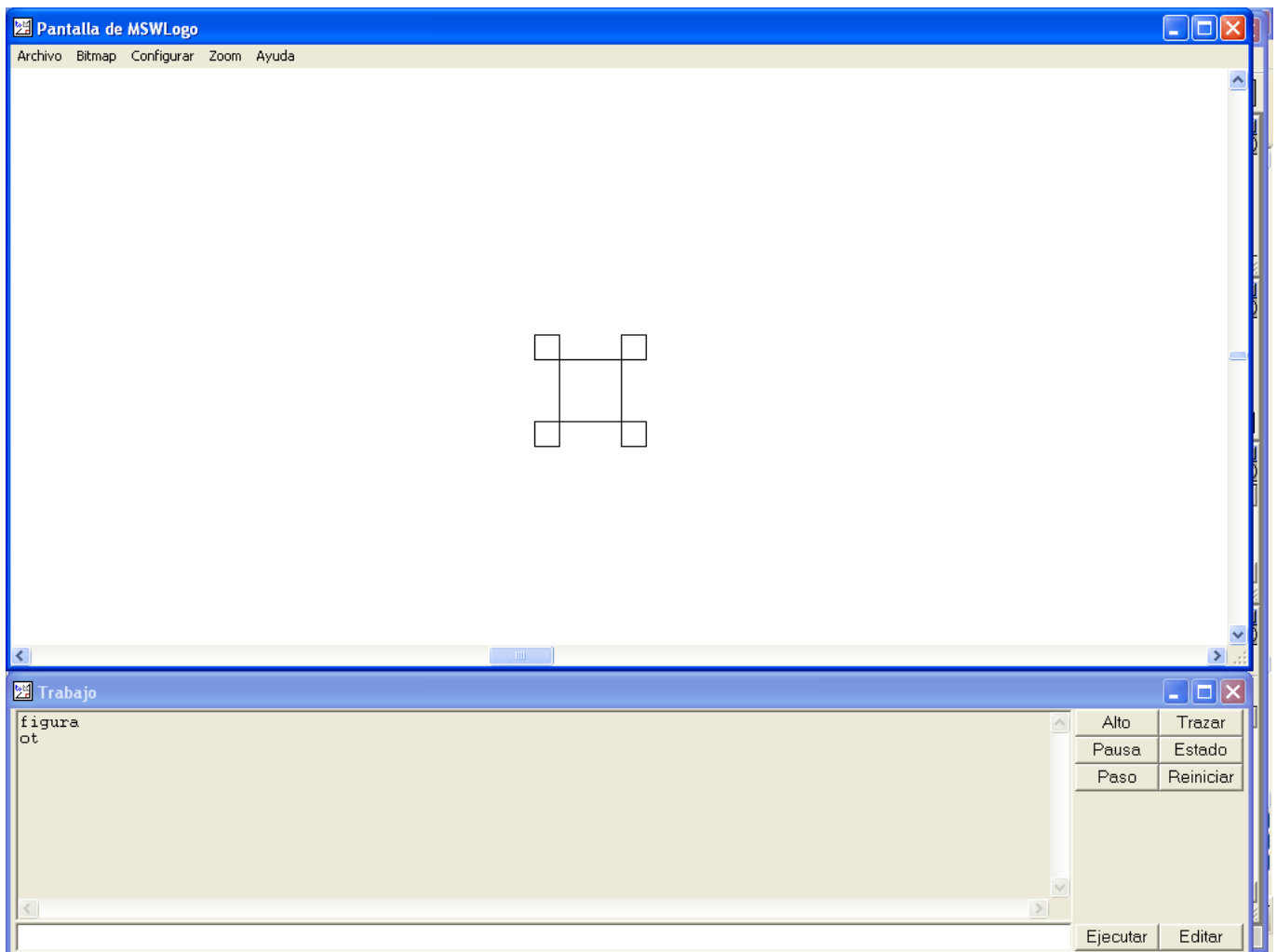
```
PARA POLI :LADO :ANGULO
AVANZA :LADO
GIRADERECHA :ANGULO
POLI :LADO :ANGULO
FIN
POLI 60 80
```

Como estarás comprobando el programa ha entrado en un ciclo infinito y está tan ocupado que no te hace caso, cuando quieras retomar el control del programa. Pulsa el botón "ALTO" situada en la parte derecha de la ventana trabajo y se detendrá la ejecución del programa.

Modularización.

Conforme vamos aprendiendo más cosas sobre la programación con MSWLogo vamos exigiéndonos más y nos planteamos problemas más complejos. Cuando esto ocurre resulta necesario ver si podemos descomponer el problema complejo en una serie de problemas sencillos o módulos.

Un ejemplo quizás nos ayude a entender esto. Si deseamos realizar la figura que se muestra (un ejemplo que aun está a nuestro alcance) la podemos descomponer en procedimientos que me dibujen figuras más sencillas y realizar después un procedimiento que integre los pequeños módulos en que hemos descompuesto nuestro problema.



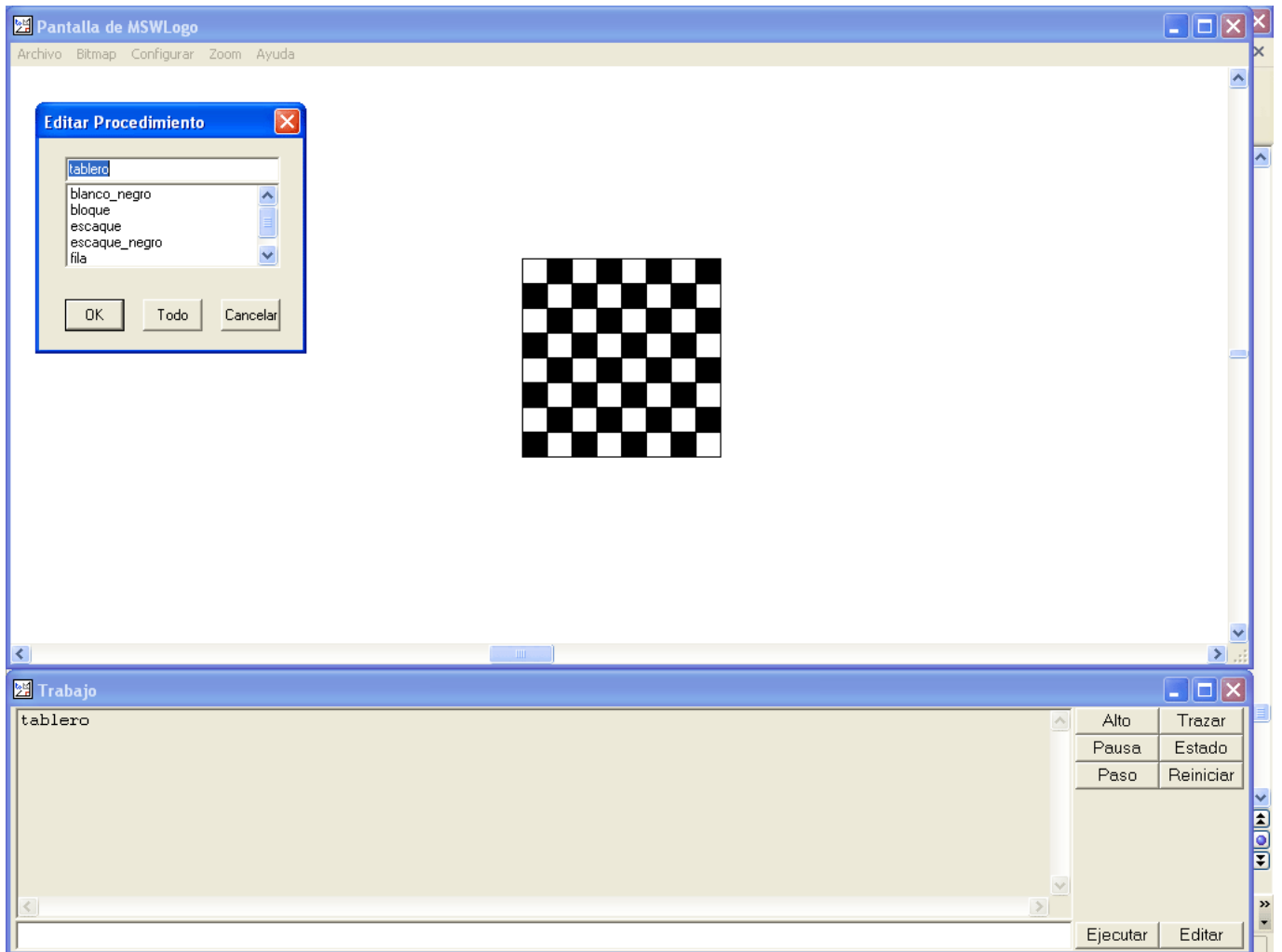
El programa podría consistir en una serie de procedimientos como los que se muestran a continuación

```

PARA FIGURA
REPITE 4 [BANDERA]
FIN
PARA BANDERA
CUADRADO 20 MASTIL
FIN
PARA MASTIL
GI 90 AV 50
FIN
PARA CUADRADO :LADO
REPITE 4[AV :LADO GD 90]
FIN
FIGURA

```

Si deseásemos dibujar un tablero de ajedrez podríamos descomponer el dibujo en otros más sencillos. Podríamos empezar definiendo procedimientos que me dibujasen un cuadrado blanco y otro negro. Continuaríamos definiendo un procedimiento que me dibujase un cuadrado blanco junto a uno negro, a continuación definiríamos el procedimiento que completase una fila y seguiríamos hasta completar el tablero. No es fácil pero si no empiezas no terminarás. ¿Te atreves?



Estructuras de control.

Condicionales.

La primitiva SI es de la forma

Si pred [órdenes]

Ejecuta la lista de instrucciones indicada en órdenes en caso de que pred sea CIERTO, en caso de que sea FALSO no hace nada.

```

PARA NUMERO :NUMERO
SI :NUMERO<0 [ESCRIBE "NEGATIVO]
FIN
NUMERO -17

```

```

PARA EDAD :EDAD
SI :EDAD<18 [ESCRIBE [MENOR DE EDAD]]
FIN
EDAD 17

```

Si deseásemos obtener respuesta tanto para el caso CIERTO, como para el caso FALSO deberíamos utilizar la orden SiSino

SiSino pred [órdenes] [órdenes2]

Ejecuta la lista de instrucciones indicada en órdenes en caso de que pred sea CIERTO. Si es FALSO se ejecutará la segunda lista, órdenes2.

Para aclarar ideas probemos estos 2 procedimientos

```

PARA EDAD :EDAD
SISINO :EDAD<18 [ESCRIBE [MENOR DE EDAD]] [ESCRIBE "MAYOR]
FIN
EDAD 17

```

Busca en la ayuda información sobre la primitiva AZAR e intenta averiguar que haría el siguiente procedimiento

```

PARA MONEDA
SISINO 1=AZAR 2 [ESCRIBE "CARA] [ES "CRUZ]
FIN
MONEDA

```

Detención de un procedimiento.

La primitiva ALTO detiene la ejecución del procedimiento en curso, devolviendo el control al procedimiento que lo ha llamado.

Comprobémoslo en el siguiente ejemplo

```

PARA CONTADOR_HASTA :NUMERO
HAZ "CUENTA 0
INCREMENTO
FIN
PARA INCREMENTO
ES :CUENTA
SI :CUENTA=:NUMERO [ALTO]
HAZ "CUENTA :CUENTA+1
INCREMENTO
FIN
CONTADOR_HASTA 10

```

Intenta escribir el procedimiento de nombre CUENTA_ATRÁS :NUMERO que escriba valores desde :NUMERO hasta 0.
