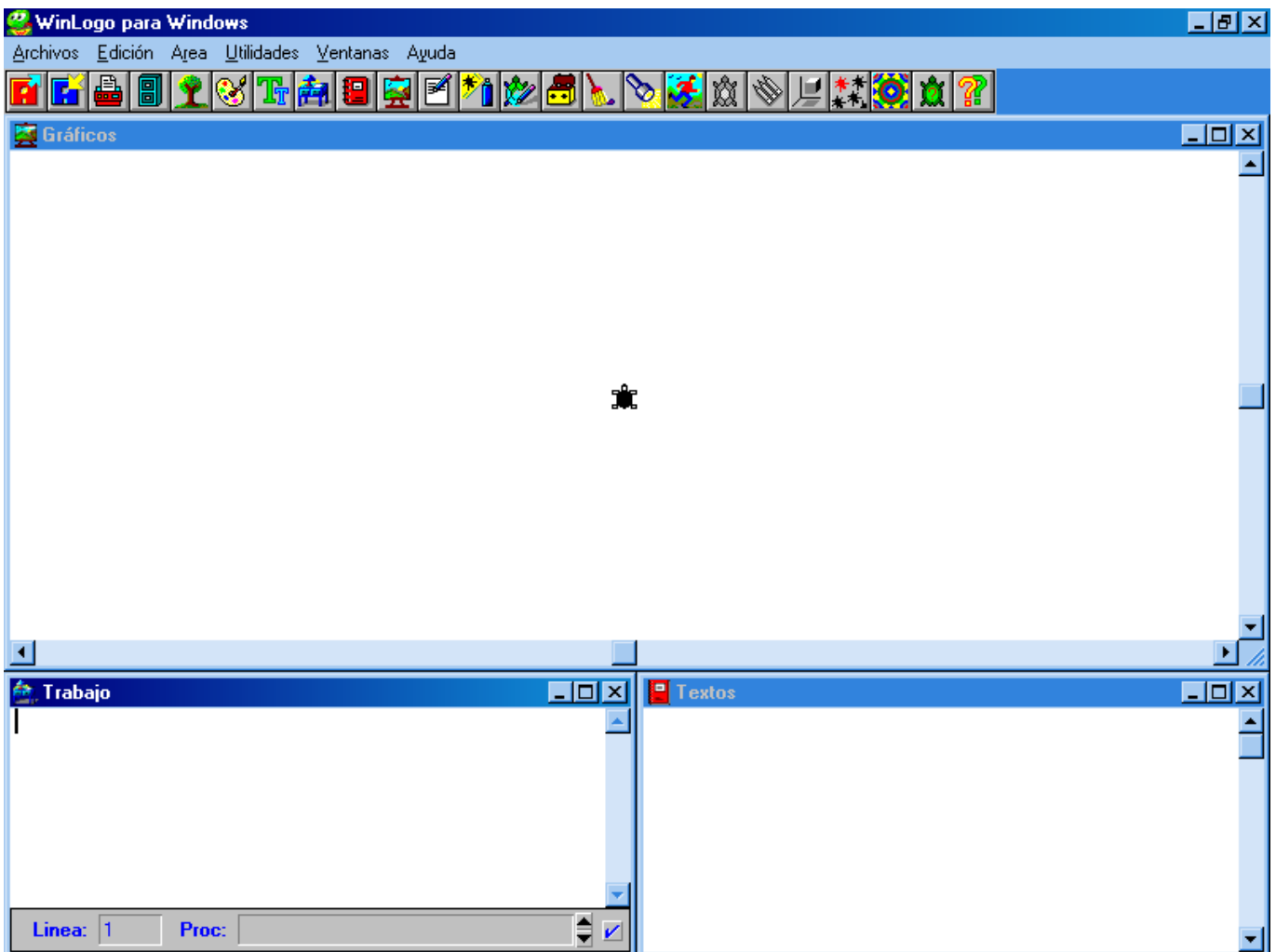


El entorno de WinLogo

Una vez arrancado el programa, nos encontraremos con una presentación como la que se muestra:



La ventana de la aplicación aparece dividida en otras tres cuyo nombre puedes localizar en la barra de título de cada una de ellas, estas son Gráficos, Trabajo y Textos.

Ventana de trabajo: en ella escribiremos las primitivas que realizarán diferentes operaciones

Ventana de gráficos: en el centro de la misma aparece una tortuga que se desplazará dejando un rastro en función de las primitivas que hayamos escrito en la ventana de trabajo.

Ventana de textos: esta ventana la utiliza el programa para mostrar diferentes tipos de información (no conoce alguna primitiva que le escribamos, no sabe como hacer algo, o el resultado de algunas primitivas cuando estas dan valores de tipo texto o numérico).

Movimiento. Primeras primitivas.

Las cosas que sabe hacer WinLogo 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 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 a la tortuga 60 unidades al tiempo que va dibujando una línea, mientras que la primitiva GIRADERECHA le hace girar sobre si misma 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. 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 WinLogo 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 ejecutabamos “procedimientos primitivos” o “primitivas” (acciones que ya sabe hacer WinLogo). Ahora le vamos a enseñar a hacer cosas que no sabe mediante “procedimientos”.

Definición de Procedimientos

Cuando se define un procedimiento a través de la primitiva PARA, WinLogo entra en modo definición de procedimiento, y permanece así hasta que se pulse INTRO en el Fin del procedimiento.

Además, la línea de estado presenta también el nombre del procedimiento que está definiendo. Cada vez que desplace el cursor por encima de un procedimiento definido, podrá ver que su nombre aparecerá dentro del recuadro.

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. Al pulsar Intro sobre esta última línea me aparecerá en la ventana de textos “Acabas de definir ... (Nombre que le hemos dado al procedimiento)”.

En caso de realizar cualquier modificación sobre un procedimiento debo ir a la línea que tiene la primitiva fin y pulsar Intro para que el ordenador reconozca los cambios realizados.

Por ejemplo, si queremos definir un procedimiento que dibuje triángulos equiláteros podríamos hacer lo siguiente

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

Al escribir el nombre del procedimiento en la ventana de trabajo

TRIANGULO

El programa me dibujará un triángulo.

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

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.

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

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

Veamoslo 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 ESC y se detendrá la ejecución del programa.

Una forma más elegante de hacer eso es insertar la siguiente línea

```
SI TECLA? [ALTO]
```

en cualquier sitio dentro del ciclo de recursividad.

Modularización.

Conforme vamos aprendiendo más cosas sobre la programación con WinLogo vamos exigiendonos 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.

The screenshot shows the WinLogo application window. The main canvas displays a square with a cross inside it. Below the canvas are two panels: 'Trabajo' and 'Textos'. The 'Trabajo' panel contains the following code:

```

GI 90 AV 50
FIN
PARA CUADRADO :LADO
REPITE 4[AV:LADO GD 90]
FIN
FIGURA
  
```

The 'Textos' panel contains the following text:

```

Acabas de definir FIGURA
Acabas de definir BANDERA
Acabas de definir MASTIL
Acabas de definir CUADRADO
  
```

At the bottom of the window, there is a status bar showing 'Linea: 14' and 'Proc: []'.

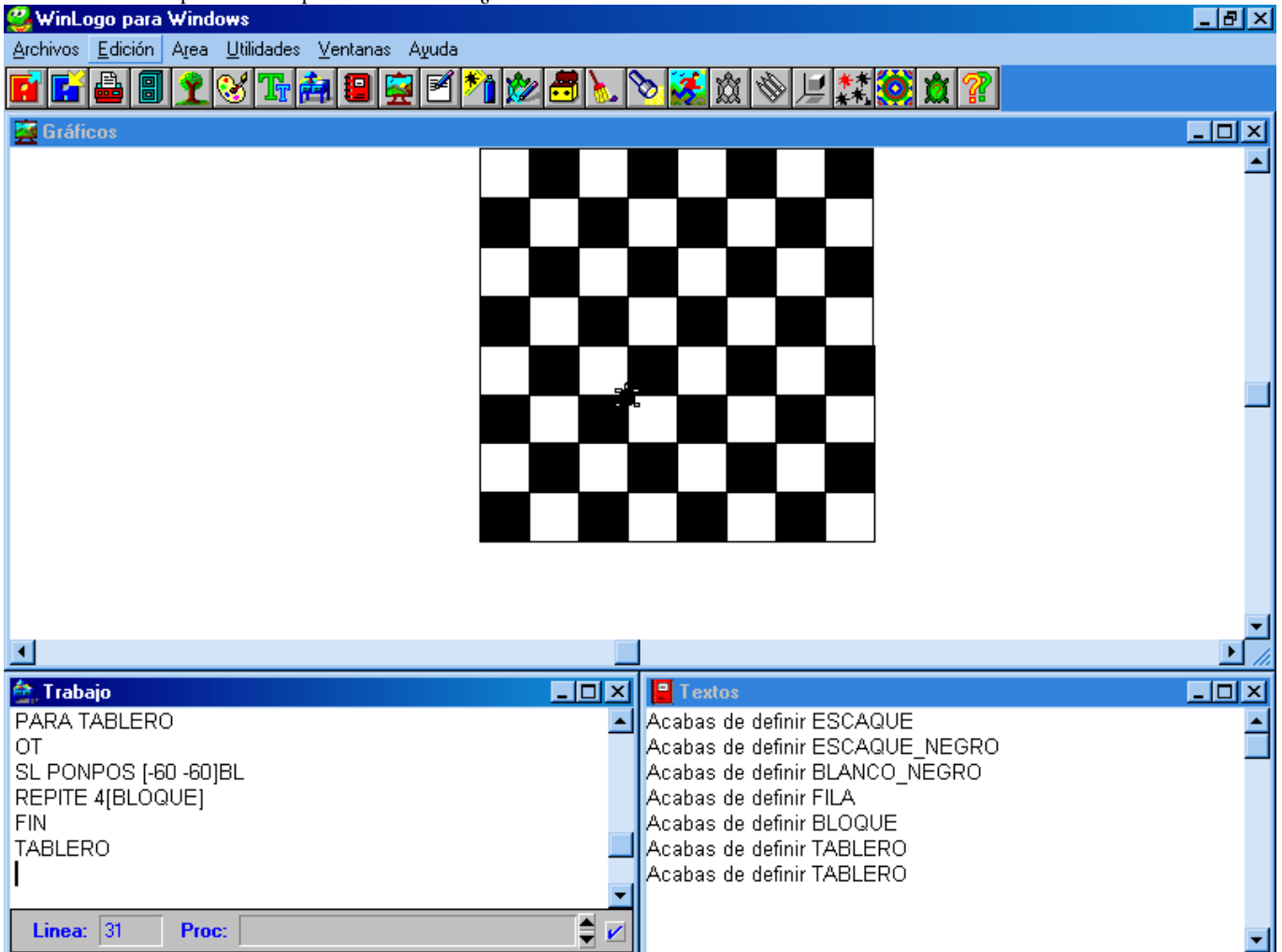
El programa podría consistir en una serie de procedimientos como los que se muestran a continuación

```

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 deseamos 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 terminas. ¿Te atreves?



Estructuras de control.

Condicionales.

Utilizamos con anterioridad

```
SI TECLA? [ALTO]
```

La primitiva SI es de la forma

```
Si pred [órdenes]
```

```
Si pred [órdenes] [órdenes2]
```

Ejecuta la lista de instrucciones indicada en órdenes en caso de que pred sea CIERTO. Si se indica una segunda lista, órdenes2, ejecutará la misma en caso de que pred sea FALSO. Si se omite la lista órdenes2, se debe pulsar INTRO después de la lista órdenes, de lo contrario se producirá un mensaje de error.

Para aclarar ideas probemos estos 2 procedimientos

```
PARA EDAD :EDAD
```

```
SI :EDAD<18 [ESCRIBE [MENOR DE EDAD]][ESCRIBE "MAYOR]
```

```
FIN
```

```
EDAD 17
```

```
PARA NUMERO :NUMERO
```

```
SI :NUMERO<0 [ESCRIBE "NEGATIVO]
```

```
FIN
```

```
NUMERO -17
```

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

```
PARA MONEDA
```

```
SI 1=AZAR 2[ESCRIBE "CARA][ES "CRUZ]
```

```
FIN
```

```
MONEDA
```

Detención de un procedimiento.

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