

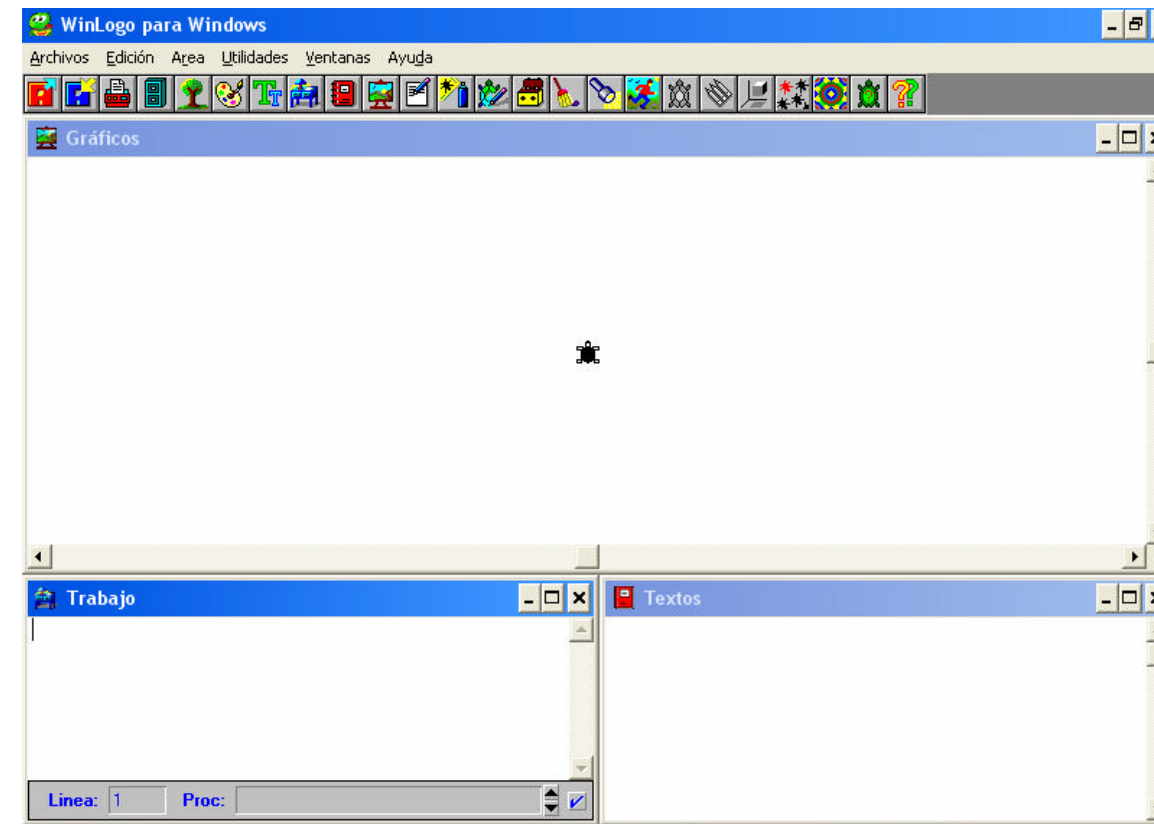
PROGRAMACION WINLOGO



1. LAS VENTANAS DE WINLOGO.

WinLogo posee varias ventanas o áreas de trabajo que pueden ser utilizadas para determinadas tareas. Estas ventanas son: Gráficos, Textos, Trabajo, Formas, Edición, Trazado, Variables.

Cuando se inicia el programa se presentan las ventanas o áreas de trabajo más importantes, que son: la ventana de Gráficos (mundo de las tortugas), la ventana de Textos (zona en la que Logo escribe sus mensajes) y ventana correspondiente al área de Trabajo (zona en la que el usuario escribe sus mensajes).



Cómo trabajar con ventanas

Las ventanas no están siempre en la misma posición: se pueden mover, cambiar de tamaño, superponer o cerrar. Solamente se puede actuar sobre las ventanas activas. Para determinar cuándo una ventana se encuentra activa es suficiente comprobar que su barra superior se encuentra resaltada. Para activar una ventana basta colocar el puntero del ratón sobre ella y pulsar su botón izquierdo; el camino indicado sirve solamente si la ventana está presente en pantalla; en caso contrario, es decir cuando la ventana no esté en pantalla, basta elegir la opción Ventanas de la barra de menú y seleccionar el nombre de la ventana que se quiere activar; la ventana así activada se superpondrá a las ventanas que aparecían antes en la pantalla. Para desactivar una ventana basta con activar una nueva. Para cerrar una



determinada ventana en primer lugar ésta debe aparecer como activa; después, la ventana puede cerrarse utilizando alguno de los procedimientos siguientes:

- ? Pulsando [Ctrl]+[F4].
- ? Haciendo doble "clic" en el recuadro superior izquierdo (botón de control) de la ventana activa.
- ? Haciendo un "clic" en el botón de control de la ventana y eligiendo la opción Cerrar.

Para mover una ventana hay que situar el puntero del ratón sobre el título de la ventana y mantener pulsado el botón izquierdo del ratón mientras la ventana se desplaza hasta la posición deseada. Una vez que la ventana ha llegado hasta el lugar deseado basta soltar el botón. Para modificar el tamaño de una ventana lo primero que hay que hacer es activarla y después situar el puntero del ratón sobre uno de sus lados hasta que aparece una doble flecha; pulsado el botón del ratón el borde de la ventana puede ser arrastrado hasta que éste alcance el tamaño deseado. Si se desea restaurar automáticamente una ventana al número y disposición iniciales, basta elegir en la barra de Menú la opción Ventanas y seleccionar, dentro de ella, el comando "Restaurar Entorno".

2. EL MÉTODO DIRECTO.

Las palabras clave que se usan para comunicarse con el entorno se llaman PRIMITIVAS. Las primitivas pueden escribirse en la ventana de TRABAJO.

Cuando el objetivo de la primitiva sea dibujar, sus resultados se muestran en el área de GRÁFICOS y, por el contrario, aparecen en el área de TEXTOS cuando la primitiva tiene por objeto un texto.

Con el ejemplo siguiente quedará más claro el funcionamiento del modo directo: Lo primero que debe hacerse es activar el área de TRABAJO; para ello se sitúa el puntero del ratón sobre algún punto de la ventana de TRABAJO y después se hace "clic" en ella con el ratón. Instantáneamente el título de esa ventana aparecerá resaltado, lo cual indica que esa zona es activa.

Si se escribe:

AVANZA 40 y se pulsa [INTRO]

La tortuga dibujará una recta de cuarenta unidades de longitud en la ventana GRÁFICOS, ya que el resultado de la primitiva AVANZA es un dibujo.

Prueba con otra primitiva, escribe ESCRIBE "HOLA y después pulsa [INTRO]. En la ventana de TEXTOS aparecerá escrita la palabra HOLA. Las órdenes de escritura se ejecutan en la zona de TEXTOS.

Si se comete un error al escribir una primitiva como por ejemplo ordenar a la primitiva una acción incompleta, Logo envía su mensaje de error a través de la ventana de TEXTOS. Para comprobarlo se tecléa: ESCRIBE "HOLA; la palabra escribe se ha escrito con dos eses; el ordenador en este caso emitirá un mensaje de error en la ventana de TEXTOS. El mensaje de error que va a parecer es: no se cómo hacer ESCRIBE.

Realiza las pruebas siguientes: escribe los textos tal y como aparecen a continuación y pulsa [INTRO] después de cada línea.

AVANZA 80

AV 80



BORRAPANTALLA
 AV 40
 GIRADERECHA 90
 AV 40 GD 90
 BP AV 60
 GIRAI ZQUI ERDA 45
 RETROCEDE 60
 GI 90 RE 56
 ESCRI BE "CASA
 ESCRI BE [LA CASA ES GRANDE]
 BP

ACTIVIDAD Nº1. REALIZAR UN CUADRADO.

Dibuja un cuadrado y un triángulo equilátero haciendo uso de las primitivas que has visto hasta ahora.

PRIMITIVA ABREVIATURA ACCIÓN

AVANZA x AV x la tortuga se desplaza x unidades hacia adelante
 RETROCEDE x RE x la tortuga se desplaza x unidades hacia atrás
 GIRADERECHA x GD x la tortuga gira a la derecha x grados
 GIRAI ZQUI ERDA x GI x la tortuga gira a la izquierda x grados
 ESCRI BE "X ES "X escribe en la ventana de textos la palabra situada detrás de las comillas
 ESCRI BE [XX XX] ES [XX XX] escribe en la ventana de textos la frase situada entre los corchetes
 BORRAPANTALLA BP se borra la pantalla de gráficos

También se pueden escribir dos o más órdenes en una misma línea. Por ejemplo:

BORRAPANTALLA AVANZA 40 GIRADERECHA 90 AVANZA 60.

Después de escribir la línea anterior en la ventana de trabajo y pulsar [I NTRO], se ejecutarán las cuatro instrucciones.

Edición en modo directo

También se puede repetir una orden directa pulsando [I NTRO] después de haber colocado el cursor en la línea en la que la orden se había ejecutado.

ACTIVIDAD GUIADA.

Escribe la siguiente orden y pulsa [I NTRO]:

GD 45

El cursor baja una línea y el programa espera a que escribas otra orden. Sube el cursor a la línea anterior y vuelve a pulsar [I NTRO]. Repite la acción varias veces, verás cómo se ejecuta como si la hubieras escrito de nuevo.

Como se ha explicado anteriormente, en una línea se pueden escribir varias órdenes seguidas. Logo las ejecuta una tras otra.

Borra la pantalla de Gráficos con la orden BP y, en una línea en blanco escribe las dos órdenes siguientes y pulsa [I NTRO]:

AV 60 GD 90

Sube el cursor a la línea anterior y pulsa [I NTRO]. Repite la operación dos veces más.

Cuando concluyas el ejercicio observa que Logo ha dibujado un cuadrado.

Esta posibilidad permite corregir un error sin necesidad de reescribir toda la línea de órdenes en la que aparecía dicho error.



Escribe y ejecuta:

HESCRIBE "ERROR"

Logo envía un mensaje a la ventana de textos el siguiente mensaje:

no sé como hacer para HESCRIBE

Sube el cursor a la línea errónea, suprime la H y pulsa [I NTRO]. Observa que en la ventana de textos Logo escribe la palabra que le habías ordenado.

Una primitiva interesante: REPITE

Merece la pena dedicarle un apartado específico a la primitiva REPITE debido a su gran utilidad en la creación de gráficos y en el micromundo de control. Esta primitiva no tiene abreviatura; su sintaxis es: REPITE n[acciones]

ACTIVIDAD GUIADA

Realiza los siguientes ejercicios y pulsa [I NTRO] al final de cada línea:

REPITE 4[AV 50 GD 90]

BP

REPITE 6 [AV 40 GD 60]

BP

REPITE 7 [AV 30 GD 360 / 7]

Cambia los dos sietes por dos nueves y pulsa [I NTRO].

Cambia los sietes por 360 y cambia AV 30 por AV 1. ¿Qué ocurre?

Escribe el siguiente conjunto de órdenes:

BP SL GI 90 AV 200 GD 90 BL

REPITE 10[AV 5 GD 90 AV 20 GD 90 AV 5 GI 90 AV 20 GI 90]

Observa que la última orden del corchete GI 90 deja a la tortuga en la orientación inicial de modo que, al ejecutarse el módulo siguiente, la tortuga esté situada en la misma dirección.

ACTIVIDAD Nº2. REALIZAR UNA CENEFA SIMILAR A ESTA.



3. MODO PROCEDIMENTAL

El lenguaje Logo permite trabajar en modo procedimental, característica que comparte con otros lenguajes de ordenador. Programar en modo procedimental consiste en enseñar procedimientos al ordenador que, una vez asimilados, interpreta como si fueran nuevas primitivas.

Los procedimientos pueden escribirse en las ventanas de trabajo y edición, aunque es preferible hacerlo en la de edición. Para ello, se pulsa el icono de Edición que aparece en la barra de menú que se superpone a la ventana de edición. Los procedimientos se escriben en la ventana de edición.

Existen dos PRIMITIVAS asociadas a cualquier procedimiento, PARA y FIN. Todos los procedimientos comienzan por la primitiva PARA y terminan con la primitiva FIN.



Escribe el siguiente procedimiento:

PARA CUADRADO
REPITE 4 [AV 60 GD 90]
FIN

Hay que tener en cuenta que en la primera línea, sólo puede escribirse la palabra que se está definiendo separada de PARA por un espacio y en la última línea sólo puede escribirse la palabra FIN. Siempre debe pulsarse [INTRO] para validar un procedimiento. El procedimiento anterior se llama CUADRADO. Para ejecutar posteriormente este procedimiento, es necesario que el ordenador lo memorice; para ello, basta pulsar el icono de INTERPRETE. En la ventana de Texto aparecerá, si el procedimiento es correcto, el siguiente mensaje: ACABAS DE DEFINIR CUADRADO. Con esta acción, el ordenador ha memorizado el procedimiento, por lo que puede ser utilizado en el futuro. Cierra la ventana de Edición situando el cursor en el cuadrado superior izquierdo y pulsa dos veces el ratón. Activa la ventana de trabajo.

Si escribes CUADRADO y pulsas [INTRO], en la ventana de gráficos se ejecutarán las órdenes asociadas a ese procedimiento (en este caso aparece cuadrado de 60 unidades de lado).

Los procedimientos recién definidos pueden ejecutarse mientras no se salga de Logo. Una vez apagado el sistema, Logo sólo recuerda sus primitivas.

Cómo archivar los procedimientos que se encuentran en el Editor.

Para no perder los procedimientos diseñados de un día para otro, Logo permite archivar en el disco los procedimientos que se encuentran visibles en la pantalla de edición. Para archivar los procedimientos pueden seguirse los siguientes caminos:

- ☞ Desde la ventana de Trabajo: Mediante la primitiva GUARDAED se almacenan todos los procedimientos en un determinado archivo. Por ejemplo, con GUARDAED "arch.log, se almacenan los procedimientos en el archivo arch. log.
- ☞ Desde la barra de menú teniendo activa la ventana de EDICIÓN: Se elige la opción GRABAR dentro del menú ARCHIVO. Cuando aparece la ventana de diálogo adjunta, se escribirá el nombre con el que se quiera guardar el conjunto de procedimientos. En este caso se elige el nombre ARCH. Conviene señalar que los archivos de procedimientos, tienen por defecto la extensión LOG.
- ☞ Teniendo activa la ventana de EDICIÓN, es posible elegir el icono GRABAR (salvar en disco) y seguir los pasos del punto anterior.

ACTIVIDAD N°3. Realizar un procedimiento TRIAN para un triángulo equilátero.

4. ALGUNAS PRIMITIVAS MÁS DEL LENGUAJE LOGO

PRIMITIVA ABREVIATURA ACCIÓN

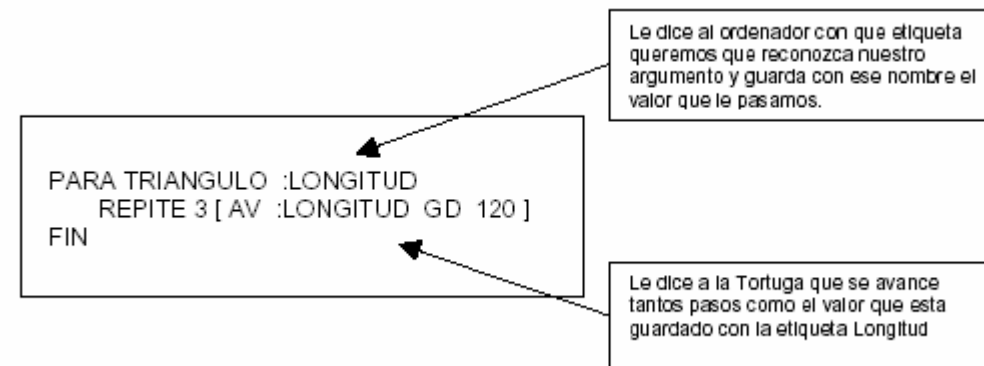
SUBELAPIZ Eleva el lápiz de la tortuga activa para que SL no deje trazo al moverse.
BAJALAPIZ BL Activa el lápiz de la tortuga y dibuja un trazo a lo largo de su trayectoria.
GUARDAED Guarda el contenido del área de edición en "archi un archivo llamado archi.
REPITE Repite el número de veces indicado la acción situada entre corchetes.
CENTRO Sitúa las tortugas activas en el centro del gráfico. En su desplazamiento al centro, las tortugas dejan rastro o no, dependiendo de si su posición es bajalapiz o subelapiz.



5. PROCEDIMIENTOS CON UNA VARIABLE.

Para crear dos triángulos de tamaño diferente a la vez necesitas clonar TRI ANGULO para producir TRI -50 y TRI -267. Para tener 10 tamaños diferentes a la vez necesitas 10 clones. Esta es sin duda una manera muy laboriosa de tener triángulos de diferentes tamaños a la vez ¿verdad?.

Logo tiene una manera mejor de hacerlo. Puedes darle al TRI ANGULO un 'argumento': una longitud variable por ejemplo, TRI ANGULO 100, TRI ANGULO 250, etc. Solo se diferencian por el número que va después de **AV**. Fíjate también que TRI ANGULO 100 se parece a AV 100 y que TRI ANGULO 250 se parece a AV 250. **ED "TRIAN** (asegúrate de tener la actividad nº 3 cargada) y cambia el procedimiento de la forma indicada aquí debajo:



ACTIVIDAD N°4. Realizar un procedimiento cenefa para conseguir distintos tamaños.

REPI TE 10[AV 5 GD 90 AV 20 GD 90 AV 5 GI 90 AV 20 GI 90]. Si introducimos esta sentencia conseguimos una cenefa, pero se trata de modificar los tramos de 20 pasos (AV 20) por tramos de longitud variable.

Experimenta con tus figuras y sus datos variables. Prueba por ejemplo la siguiente figura.

```

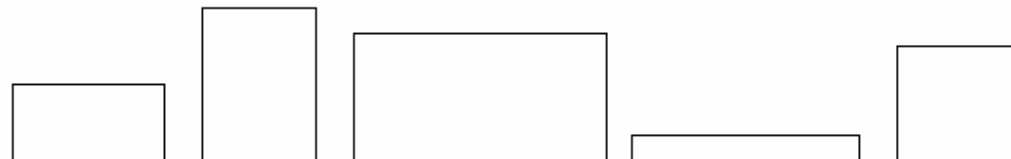
PARA FIGURA
  REPI TE 3 [ TRI ANGULO 100 GD 120 ]
FIN
  
```

Observa qué otras interesantes figuras puedes crear usando TRI ANGULO, CUADRADO, HEXAGONO, OCTOGONO, etc. con **diferentes REPI TE datos variables y diferentes valores para el 'ángulo de giro'**

6. PROCEDIMIENTOS CON VARIAS VARIABLES

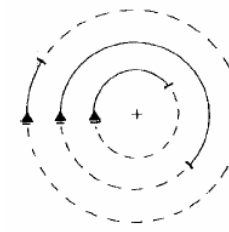
Antes de poder dibujar figuras realmente interesantes tenemos que enseñarle a nuestra tortuga dos nuevos procedimientos que aun no conoce: el rectángulo y el arco.

ACTIVIDAD N°5. Enseña a tu tortuga a dibujar todos los rectángulos que ves a continuación mediante un procedimiento de dos variables: base y altura.



ACTIVIDAD N°6. Enseña a tu tortuga a dibujar trozos de arcos como se muestra en la figura basándote en el procedimiento de dibujar círculos (estudiado en la actividad guiada de la página 4)

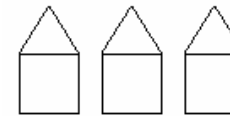
Fíjate que necesitamos dos variables: el ángulo del arco y el radio de la circunferencia.



ACTIVIDAD N°7. Ahora crea un procedimiento para construir una CASA.




ACTIVIDAD N°8. Ahora crea un procedimiento para construir un PUEBLO que permita el número de casas que deseemos. (Nos hace falta un procedimiento HUECO para ir de una casa a otra sin dejar marca)










Uniendo figuras simples.

Empieza dibujando una TORRE. Puedes necesitar el RECTANGULO y el TRIANGULO. Clona la CASA y úsala de modelo para realizar la TORRE.

	PARA TORRE
	FIN

	PARA CASTILLO
	FIN

	PETALO
	
	FLOR

	ARCO
	RAMA
	LLAMA
	SOL

**ACTIVIDAD DE AMPLIACIÓN.
MUESTRA CADA FIGURA
CONSEGUIDA AL PROFESOR
PARA AUMENTAR TU
CALIFICACIÓN.**



7. USO DE VARIABLES GLOBALES.

Las variables pueden ser comparadas con cajas a las que se les pone una etiqueta (nombre) y en cuyo interior existe un contenido. Una variable, por ejemplo, puede llamarse HOLA y contener el número 34; otra puede llamarse BARCO y contener la palabra chocolate. De este modo, al ejecutar la orden de escribir el contenido de HOLA, el ordenador escribe 34, de la misma manera que si se le mandase escribir el contenido de BARCO escribiría chocolate.

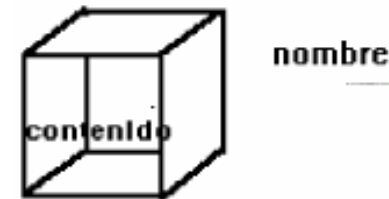
Las variables en el modo directo:

Sitúate en la ventana de trabajo.

Escribe:

HAZ "BARCO "chocolate

Pulsa [INTRO].



Mediante la instrucción anterior se ha creado la variable BARCO y se le ha asignado como contenido la palabra chocolate.

Escribe :

ESCRIBE :BARCO

Pulsa [INTRO].

En la ventana de textos, aparece escrito el contenido de la variable BARCO, en este caso chocolate. Conviene destacar que para conocer el contenido de una variable, es necesario escribir la instrucción **ESCRIBE : "nombrevar**, donde nombrevar es el nombre de la variable:

Prueba el siguiente ejemplo:

HAZ "h 66

ESCRIBE :h+9

En este caso en la ventana de textos aparecerá el número 75, que es el resultado de sumar 66, contenido de la variable h, al número 9.

HAZ "NOMBRE contenido

- ? El nombre que se da a la variable debe ser un conjunto de caracteres sin espacios en blanco y debe ir precedido por comillas. El contenido de la variable puede ser:
- ? Un número
- ? El resultado de una operación
- ? Una palabra que irá precedida de comillas
- ? Una lista entre corchetes
- ? Una serie de datos introducidos desde teclado durante la ejecución del programa por medio de ciertas primitivas.
- ? El contenido de una variable aparece cuando se escribe precedida por el símbolo **:nombre.**

No es posible crear una variable global ni en la primera línea de un procedimiento, ni en la última.

Sirva como ejemplo el siguiente procedimiento:

PARA CUAGLO

BP BT

ES [Escribe la longitud del lado de un cuadrado]

HAZ "LADO LEEPALABRA

REPI TE 4 [AV :LADO GD 90]

FIN



```

PARA SALUDO
ES [¿Cómo te llamas?]
HAZ "NOMBRE LEEPALABRA
ES FRASE [Encantado de conocerte] :NOMBRE
FIN

```

```

PARA CUBO
BP BT
ES [Este programa calcula el cubo de los números ]
ES [¿De qué cifra deseas el cubo?]
HAZ "CI FRA LEECAR
HAZ "CUBO :CI FRA * :CI FRA * :CI FRA
ES (FRASE [El cubo de ] :CI FRA [ es ] :CUBO)
FIN

```

```

LEECAR (LC)
Lee el primer carácter de un archivo o la primera
pulsación desde el teclado sin necesidad de pulsar
[INTRO].

LEECARS num (LCS num)
Lee los num primeros caracteres pulsados desde el
teclado o leídos de un archivo

LEELISTA (LL)
Lee una lista de caracteres introducidos desde el
teclado o leídos desde un archivo

LEEPALABRA (LP)
Lee la primera palabra introducida desde el teclado o
leída desde un fichero

```

A continuación tienes tres ejemplos: el primero lee solo el primer carácter del teclado, el segundo una palabra, y el tercero una frase.

```

PARA lectura_caracteres1
ES [¿Quién descubrió América?:]
HAZ "H LC ; se asigna a la variable H el 1 carácter escrito por teclado er
ES :H
FIN

```

```

PARA lectura_caracteres2
ES [¿Quién descubrió América?:]
HAZ "H LCS 3 ; se asigna a la variable H los tres primeros caracteres escritos
por teclado
ES :H
FIN

```

```

PARA lectura_caracteres3
ES [¿Quién descubrió América?:]
HAZ "H LP ; asignamos a la variable H la primer palabra escrita por teclado
ES :H
FIN

```

```

PARA lectura_caracteres4
ES [¿Quién descubrió América?:]
HAZ "H LL ; asignamos a la variable H todo el texto escrito por teclado
ES :H
FIN

```

8. RECURSIVIDAD.

Se llama recursividad a la posibilidad que ofrece un programa diseñado en Logo de llamarse a sí mismo en un bucle sin fin. Un sencillo ejemplo lo constituye el siguiente programa, que dibuja una circunferencia de modo recursivo:

```

PARA CI REC
AV 1
GD 1
CI REC
FIN

```



Un primer problema por resolver en un procedimiento recursivo, es encontrar la forma de abandonar su ejecución infinita de procedimiento sin pulsar la tecla [ESC]. se puede abandonar dicha ejecución mediante la primitiva TECLA? en una condición.

Sentencias condicionales

Logo puede "valorar" el contenido de una condición y "decidir" entre dos opciones. Para dejar claros estos conceptos se utilizará la primitiva TECLA?, que devuelve CIERTO o FALSO según se haya pulsado o no una tecla.

El programa CI REC queda controlado al anteponer el condicional SI a la primitiva TECLA?

```

PARA ARCO
SI TECLA? [ALTO]
AV 1
GD 1
ARCO
FIN

```

Este programa va dibujando un arco con avances y giros hasta que se pulse una tecla.

Este programa va dibujando un arco con avances y giros hasta que se pulse una tecla. La sentencia condicional se expresa con la primitiva SI y tiene varias sintaxis; las dos más usadas son:

SI condición [acción 1] [acción 2]

Esta sentencia significa que, si se cumple la condición, se ejecutan las acciones contenidas en el primer corchete y, en caso contrario, las incluidas en el segundo. En ambos casos, una vez realizadas todas las acciones incluidas en uno de los corchetes, el programa ejecuta la línea siguiente del programa.

SI condición [acción]

En este caso, si se cumple la condición, se ejecutan las acciones contenidas en el corchete; después el programa pasa a la línea siguiente; si no se cumple la condición, el programa pasa a la línea siguiente.

Control de la recursividad por medio de un contador

Ejemplo:

```

PARA CONTADOR :N
SI :N<0 [ESCRIBE "GOL ALTO]
ES :N
CONTADOR :N-1
FIN

```

El programa contiene una variable local :N que sólo sirve para escribir su contenido cada vez que el programa se ejecuta. El programa se llama a sí mismo y hace disminuir el valor de la variable en cada paso. La sentencia condicional de la primera línea controla la ruptura del bucle en el momento en que se llega a un valor negativo, se escribe la palabra GOL y se detiene el programa. Obsérvese que la comparación numérica entre el contenido de la variable y la constante no se ha hecho con el signo = sino con el signo < ya que, si el incremento se hiciera con decimales (CONTADOR :N-0, 3), nunca se cumpliría la condición y el bucle sería indefinido.



Vamos a realizar un procedimiento que genere un cuadrado de lado 100 y se rellene de negro. Llama a este procedimiento CUADRONEGRO.

PARA CUADRONEGRO
REPITE 4 [AV 100 GD 90]
SL GD 45 AV 50 BL
RELLENA
SL AV -50 GI 45
FIN

La orden Rellena siempre aplica el color del lápiz de ese momento.

Aquí tienes una forma de provocar una intermitencia entre un cuadro negro y una pantalla blanca, de forma que el cuadro negro permanezca 2 segundos y la pantalla blanca 4 seg.

PARA CONTROL1
BP ESPERA 400
CUADRONEGRO ESPERA 200
CONTROL1
FIN

Se denominará al procedimiento CONTROL1. Para que el ciclo se realice ininterrumpidamente se utilizará la recursividad. Se colocará como última orden de este procedimiento la orden CONTROL1, con lo que se inicia de nuevo el procedimiento al llegar a este punto. Se denomina **recursividad** a la introducción, en la definición de un procedimiento de una llamada a sí mismo.

Ahora deseamos que nuestro CUADRONEGRO pase de negro a rojo cada 4 segundos por lo que será necesario realizar un procedimiento CUADROROJO y provocar recursividad. Para temporizar cada estado se utilizará la primitiva ESPERA ya descrita: ESPERA 400 y ESPERA 200.

ACTIVIDAD N°9. Ahora crea un procedimiento CUADRONEGRO, otro CUADROROJO y provoca una intermitencia entre uno y otro de 2 segundos.

9. VIRTUALEDS

Es una utilidad creada por un profesor de Tecnología y su utilidad consiste es poder simular tu programa de encendidos y apagados de leds.

Admite solo la orden Enviadato n°, de forma que si enviamos un 0 apaga todos los diodos y un 255 encendería todos los diodos. Con esto vamos a realizar una serie de prácticas para poder conseguir que funcione nuestro semáforo.

ACTIVIDAD N°10 Haz un programa que encienda los leds de uno en uno de forma ascendente y luego descendente.

ACTIVIDAD N°11 Haz un programa que encienda los cuatro primeros, y después apague estos cuatro y encienda los otros cuatro.

ACTIVIDAD N°12 Simula el funcionamiento de un semáforo de forma que el orden de las luces sea R1 A1 V1 R2 A2 V2.



10. SEMÁFORO.

A continuación vamos ya a realizar un semáforo en la pantalla de logo. Vamos a crearlo utilizando varios procedimientos.

Empezaremos por crear un rectángulo de color verde:

```
Para rectangulo
PonCL 4 ;color verde
Repite 2 [av 175 gd 90 av 80 gd 90] ;rectángulo de 175*80
GD 45 Av 10 Rellena Av -10 GI 45 ;tortuga dentro, rellenar y salir
GD 90 Av 40 GI 90 ;para dejar la tortuga en medio y
Fin ;dejarla mirando hacia arriba
```



A continuación se definen los procedimientos para crear la luz roja, la verde y la ámbar. Observa que los procedimientos son exactamente iguales, excepto en el color y en la altura a la que realizará el círculo.

```
Para rojo
Poncl 10 ;color rojo
SL Av 150 GD 90 BL ; SL, avanza, gira para centrar circulo, BL
Repite 360 [av 0.25 gd 1] ;circulo
GD 90 ;gira para mirar hacia abajo
Av 5 Rellena SL Av 145 GI 180 BL ;dentro rellena, SL baja y mira hacia arriba, BL
Fin
```

Ahora vosotros hacéis verde y ámbar teniendo en cuenta que la altura para hacer los círculos ámbar y verde, deben ser 100 y 50 respectivamente. A continuación ejecuta todos los procedimientos creados y si todo está bien tendremos el rectángulo verde con los tres círculos.

A continuación necesitamos los procedimientos que dibujen los círculos negros para que solo veamos encendido o el rojo o el verde o el ámbar.



Logo rellena el área donde la tortuga se encuentra encerrada, siempre que el lápiz sea del mismo color de las líneas que encierran la tortuga.

Podemos observar como negroverde pinta de color negro el círculo verde cuando están iluminadas las luces ámbar o roja. Es necesario por lo tanto crear también los procedimientos negroambar y negro rojo. Estos procedimientos serán exactamente iguales a los ya creados rojo, ámbar y verde con el único cambio del color que ha pasado a ser negro.

```
Para negroverde
Poncl 1
SL Av 50 GD 90 BL
Repite 360 [av 0.25 gd 1]
GD 90
Av 5 Rellena SL Av 45 GI 180 BL
Fin
```



Una vez creados negroverde, negro rojo y negroámbra prueba que si los ejecutas tus círculos se vuelven de color negro.

Ahora ya solo queda hacer un procedimiento que se utilizará para dibujar los círculos en su posición y en el tiempo necesario.

Para luces	;para ocultar la tortuga y conseguir rápida ejecución del programa
Verde	;se enciende el verde y se apagan las otras dos luces
negro rojo	
negroámbra	
Espera 600	;6 segundos para el verde
ambar	;a los 6 segundos se enciende el ambar y se ponen en negro
negro rojo	;los otros dos
negroverde	
Espera 100	; al ámbra solo le he dado 1 segundo
rojo	;pinta el círculo rojo
negroámbra	; los otros dos en negro
negroverde	
Espera 600	;le ha dado 6 segundos al rojo
luces	
Fin	

Ya solo falta dibujar el rectángulo y llamar al procedimiento luces

```

Para semaforo
OT
rectangulo
luces
Fin

```

ACTIVIDAD Nº13 Realiza el semáforo que se ha explicado en el este apartado.

ACTIVIDAD Nº14 Realiza dos semáforos de forma que su cambio de luces esté coordinado como si se tratase de un cruce de dos calles con circulación en una sola dirección

ACTIVIDAD Nº15 Realiza una modificación a la actividad anterior para que solo deje de funcionar el procedimiento al recibir del teclado la palabra "parar"



A continuación vamos a describir como podría ser el cambio necesario para que nuestro programa cambie cuando reconozca una determinada palabra del teclado. En el ejemplo que se propone el programa pasa del procedimiento semáforo al procedimiento averiado cuando se pulsa "averia" y se paran los semáforos al pulsar "parar".

A continuación se presentan las modificaciones necesarias a realizar en tus procedimientos.

Para semaforo

OT

rectangulo

Haz "palabra "inicio" ;inicializa la variable y le da un contenido para no tener ninguna palabra guardada

luces ;la variable palabra contendrá la lectura de teclado

Fin

Para luces

;Aquí insertamos todo el programa luces con la combinación de los tres semáforos

Espera 600

Si Tecla? [haz "palabra LP] ;si se pulsa una tecla leemos la palabra

Si :palabra = "averia [averiado] ; si la palabra es averia nos vamos al proc averiado

Si No :palabra = "parar [luces] ;si no es averia a lo mejor es parar y si no continuo

Fin

Para averiado

;Introducir la secuencia del intermitente

Si Tecla? [haz "h leecar alto] ;si se pulsa una tecla la leemos y volvemos a semáforo

Averiado ;si no se pulsa continuamos en averiado

Fin

