



TECNOLOGÍAS

4º de ESO

Curso 2012-2013

I.E.S. Los Albares (Cieza)

Departamento de Tecnología

Septiembre 2012

Visita el blog de Tecnología:

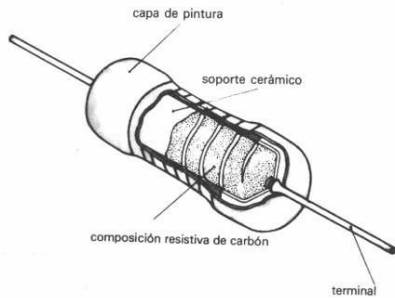
<http://davidlopezmorte.wordpress.com>

Encontrarás materiales, recursos, trabajos realizados, programación del curso, etc.

RESISTENCIAS.

Se define la **resistencia** de una sustancia como la oposición que ese material ofrece al paso de los electrones. La resistencia de un material se mide en **ohmios, Ω** , (1Ω se corresponde aproximadamente con la resistencia de un hilo de cobre de 55 m. de longitud y 1mm^2 de sección). En general, la resistencia de un material filiforme es tanto mayor cuanto mayor sea su longitud y cuanto menor sea su sección.

En un circuito electrónico las **resistencias fijas** se representan por el símbolo de la figura:



Externamente llevan unas tiras de colores que indican su valor. Las dos primeras bandas se sustituyen por dos números, según su color. La tercera indica la cifra por la que debemos multiplicar la cantidad anterior (el número de ceros que debemos añadir). La cuarta es la tolerancia, es decir el margen de error sobre el valor obtenido.



	1ª cifra	2ª cifra	3ª cifra (x)	Tolerancia
Negro	0	0	1	
Marrón	1	1	10	1%
Rojo	2	2	100	2%
Naranja	3	3	1000	
Amarillo	4	4	10000	
Verde	5	5	100000	
Azul	6	6	1000000	
Violeta	7	7	10000000	
Gris	8	8	100000000	
Blanco	9	9	1000000000	
Oro			0'1	5%
Plata			0'01	10%
Sin color				20%

Los valores normalizados para tolerancias del 5% y 10% son los siguientes:

5%) 10 11 12 15 16 18 20 22 24 27 30 33 36 39 43 47 51 56 62 68 75 82 91 100

10%) 10 12 15 18 22 27 33 39 47 56 68 82 100

A continuación vemos como se obtiene el valor de una resistencia de colores marrón-verde-naranja:

$$\begin{array}{ccc} \text{marrón} - \text{verde} - \text{naranja} \\ \Downarrow \quad \Downarrow \quad \Downarrow \\ 1 \quad 5 \quad 000 = 15000 = 15\text{K}\Omega \end{array}$$

Ahora, hagamos el proceso inverso, sacar los colores de una resistencia de $1\text{M}\Omega$

$$\begin{array}{ccc} 1\text{M}\Omega = 1 \quad 0 \quad 00000 \\ \Downarrow \quad \Downarrow \quad \Downarrow \\ \text{Marrón} - \text{negro} - \text{verde} \end{array}$$

Una **resistencia ajustable** o **potenciómetro** es una resistencia cuyo valor podemos modificar moviendo su eje o cursor. Entre los extremos del potenciómetro el valor siempre es el mismo; pero entre un extremo y el punto intermedio tendremos una resistencia variable desde 0 al valor especificado. Su símbolo es el de la figura adjunta:



Una **LDR** es una resistencia cuyo valor depende de la luz que incida sobre ella. A mayor luz menor resistencia y viceversa.



Las resistencias **NTC** y **PTC** cambian su valor en función de la temperatura. En el primer caso, NTC, la resistencia disminuye al aumentar la temperatura, en las resistencias PTC su valor aumenta al aumentar la temperatura.



Para **medir el valor de una resistencia con un polímetro** tenemos que dar los siguientes pasos:

1. Seleccionar la escala adecuada con el mando central. Los números que aparecen en esas escalas indican el valor máximo que se puede medir.
2. Conectar la sonda negra en el Terminal COM y la roja en V/ Ω .
3. Tocar, con las puntas de prueba, los extremos de la resistencia a medir.



ACTIVIDAD 1.- RESISTENCIAS

1.1.- Indicar el valor de la resistencia correspondiente a los siguientes códigos de colores:

Naranja – Naranja – Naranja

Negro – Marrón – Gris

Amarillo - Rojo – Amarillo

Violeta – Gris – Amarillo

Rojo – Rojo – Rojo

Marrón – Negro – Rojo

1.2.- Indicar el código de colores correspondiente a las siguientes resistencias:

5'6KΩ

330 Ω

1'2 KΩ

100 Ω

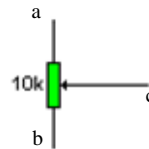
120 KΩ

4'8 KΩ

1.3. Coger seis resistencias de las que hay en el material práctico de la asignatura, obtener su valor a partir de los colores y comprobarlo con el polímetro. Rellenar la tabla adjunta:

COLORES	VALOR	MEDICIÓN

1.4.- Coger un potenciómetro de 10kΩ y realizar las mediciones indicadas en la tabla. Para mover el cursor utilizar la punta de un destornillador.



	Resistencia a-b	Resistencia a-c	Resistencia b-c
Cursor en el extremo izquierdo			
Cursor en el extremo derecho			
Cursor en el centro			
Cursor en otra posición (izq)			
Cursor en otra posición (der)			

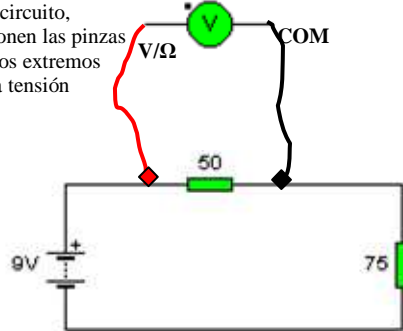
1.5.- Medir con el polímetro la resistencia que ofrece una LDR cuando incide luz sobre ella y cuando no.

	Resist. con mucha luz	Resist. con menos luz	Resist. sin luz
LDR			

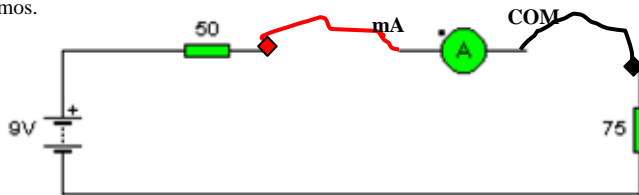
INSTRUMENTOS DE MEDIDA. EL POLÍMETRO.

Vamos a ver, con unos ejemplos, la forma de conectar un voltímetro y un amperímetro. En el circuito adjunto se muestra como se puede medir el voltaje entre los extremos de la resistencia de 50Ω y la intensidad de la corriente que la recorre:

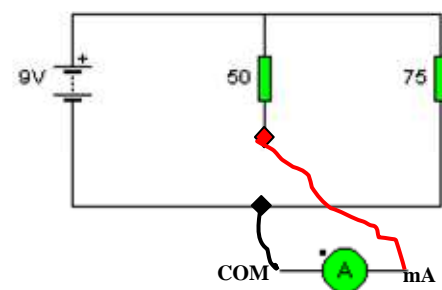
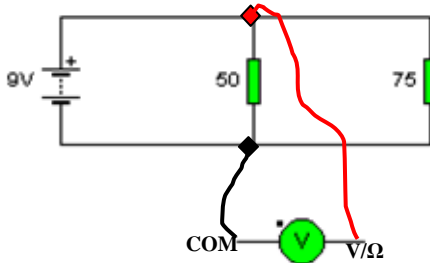
Para medir el voltaje no hace falta modificar el circuito, simplemente se ponen las pinzas del polímetro en los extremos del elemento cuya tensión queremos medir.



Para medir la corriente hay que abrir el circuito e intercalar el polímetro en los extremos.

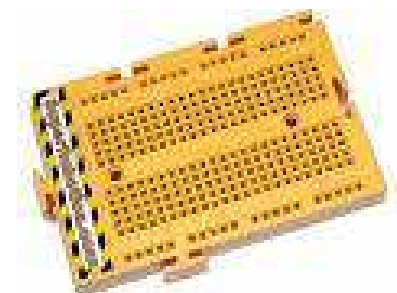
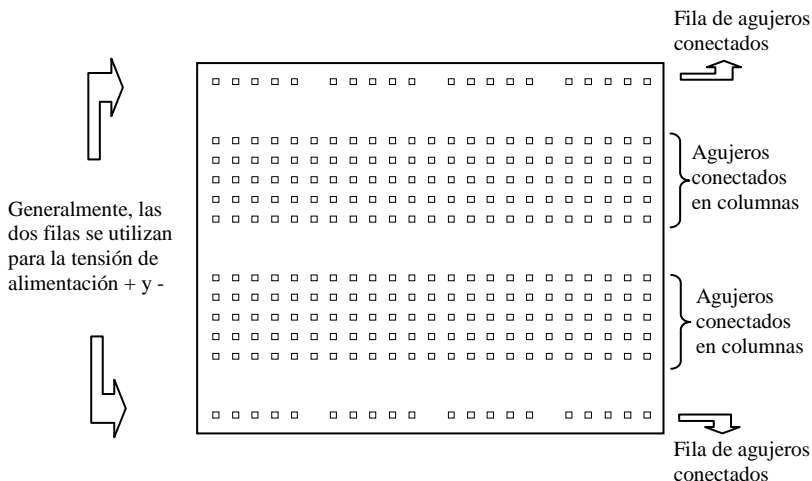


En este caso se trata de un circuito en paralelo. También medimos el voltaje y la intensidad de la corriente en la resistencia de 50Ω



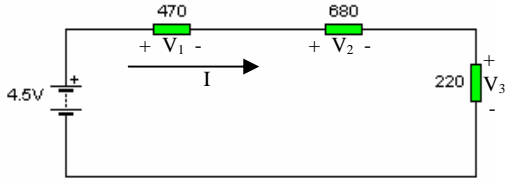
PLACAS PARA MONTAJES DE PRUEBA.

Permiten montar y desmontar con facilidad circuitos electrónicos, sin tener que soldar de manera definitiva sus elementos. Están formadas por una serie de agujeros, conectados tal como se muestra en la figura, en los que se insertan las patillas de los componentes del circuito; ayudándonos, si es necesario, de cables rígidos podemos conectar los componentes siguiendo el esquema que se pretenda realizar.

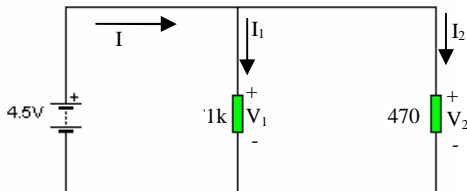
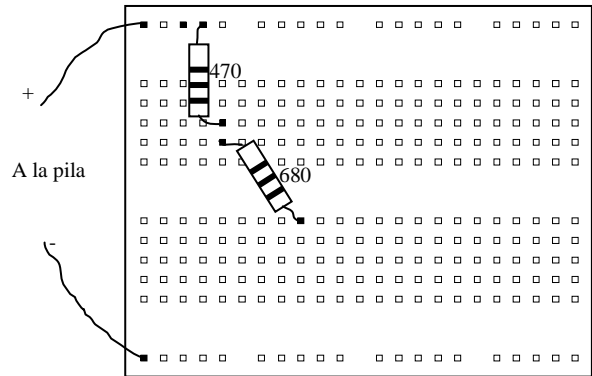


ACTIVIDAD 2.- CIRCUITOS SERIE Y PARALELO.

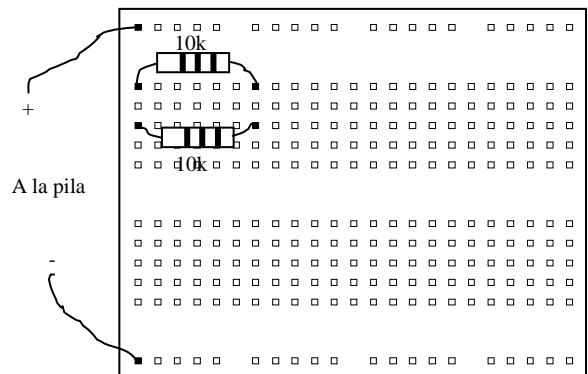
2.1.- Resolver los circuitos adjuntos, calculando la intensidad de la corriente que recorre cada uno de sus elementos y el voltaje entre sus extremos. Comprobar los resultados realizando las mediciones oportunas con el polímetro y anotarlos resultados en la tabla (indicar la escala utilizada en el polímetro para cada medición).



	I	V ₁	V ₂	V ₃
Cálculo				
Medición				
Escala				

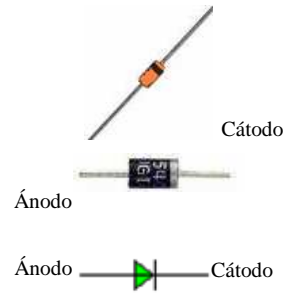


	I	I ₁	I ₂	V ₁	V ₂
Cálculo					
Medición					
Escala					

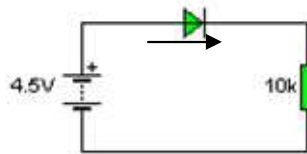


DIODOS

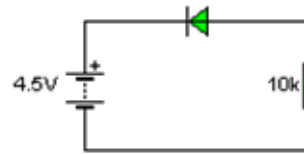
El **diodo** es un dispositivo de dos terminales, denominados ánodo y cátodo, que tiene la propiedad de permitir el paso de la corriente eléctrica sólo si va en la dirección ánodo-cátodo (la que indica el vértice del triángulo de su símbolo). En estas condiciones, es decir, cuando un diodo conduce, el voltaje entre sus extremos tiene un valor fijo de aproximadamente 0'7v y la resistencia que ofrece al paso de la corriente es prácticamente nula. Puesto que la corriente eléctrica va del polo positivo al negativo, también podemos decir que un diodo conduce cuando el ánodo es positivo respecto del cátodo.



En este circuito, el positivo de la pila va al ánodo, por tanto, la corriente va del ánodo al cátodo, el diodo conduce y hay corriente en el circuito

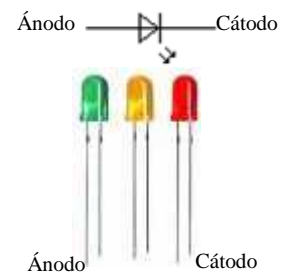


Al colocar el diodo al revés, el positivo de la pila va al cátodo, la corriente tendría que ir del cátodo al ánodo, pero el diodo no lo permite, no habiendo, por tanto, corriente en el circuito.

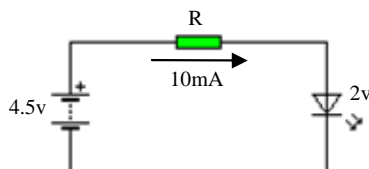


Los diodos se construyen a partir de un material denominado silicio. Añadiéndole pequeñas cantidades de otros elementos (Boro, Arsénico,...), se consiguen crear zonas donde hay muchos electrones libres (llamadas zonas N) y otras donde faltan muchos electrones (llamadas zonas P). Un diodo está formado por una zona P (el ánodo) unida a una zona N (el cátodo).

Existen unos diodos especiales que emiten luz cuando pasa la corriente denominados diodos LED. En la figura adjunta podemos ver su símbolo y tres diodos de colores verde, amarillo y rojo. Cuando un diodo led está en estado de conducción el voltaje entre sus extremos es algo superior al de un diodo semiconductor, cercano a los 2v. Generalmente, la intensidad de la corriente que debe atravesar un led para que produzca un efecto luminoso adecuado suele estar comprendida entre 5 y 20mA. Para controlar esa corriente hay que colocar en serie con el diodo led una resistencia limitadora, cuyo valor se calcula tal como se muestra en el ejemplo siguiente:



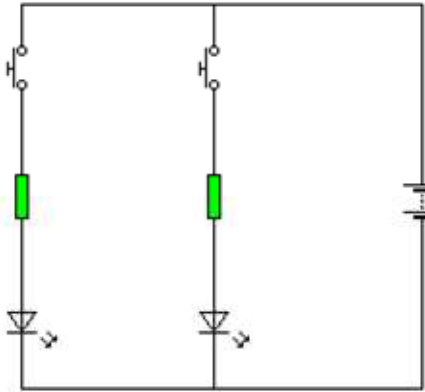
Supongamos que disponemos de una pila de 4'5v y queremos hacer pasar por el led una corriente de 10mA. Para calcular la resistencia limitadora aplicamos la ley de Ohm ($R=V/I$), por tanto, tendremos que dividir el voltaje existente entre los extremos de la resistencia por la intensidad de la corriente. Hay que tener en cuenta que dicho voltaje será la diferencia entre el voltaje de la pila (4'5v) y el que habrá en el led cuando conduzca (aproximadamente 2v). En definitiva:



$$R = \frac{V}{I} = \frac{4'5 - 2}{10} = \frac{2'5v}{10mA} = 0'25 K\Omega = 250\Omega$$

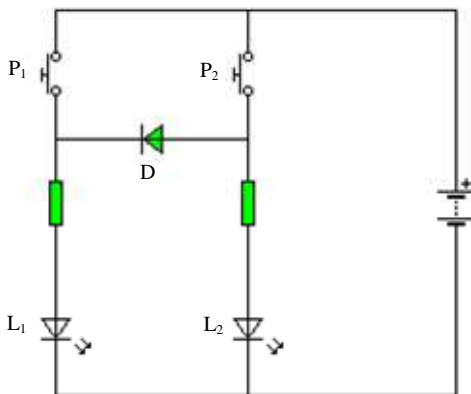
ACTIVIDAD 3.- DIODOS.

3.1.- Montar el circuito de la figura. Previamente calcular el valor de las resistencias limitadoras para que la corriente en los diodos led sea de 19 mA. Comprobar el encendido y apagado de los led al accionar los pulsadores correspondientes. Cambiar de posición el ánodo y el cátodo de algún diodo led y comprobar si, en esas condiciones, se enciende.



Cálculo de las resistencias:

3.2.- Modificar el circuito anterior añadiendo un diodo semiconductor, tal como se muestra en la figura. Comprobar en que condiciones se encienden los diodos led.



Explicar el funcionamiento del circuito:

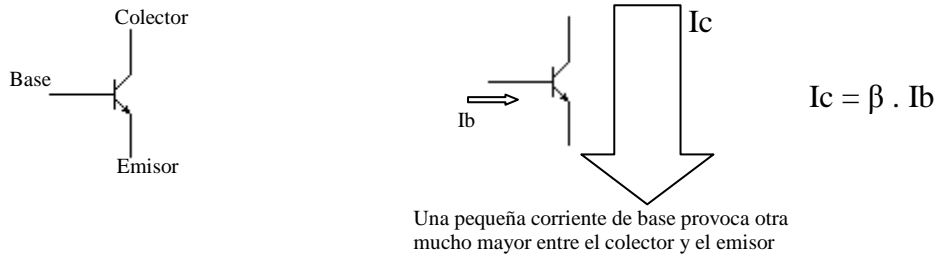
3.3.- Explicar que sucedería si cambiamos de posición los extremos del diodo semiconductor.

3.4.- Medir el voltaje entre los extremos de un diodo led cuando está encendido y del diodo semiconductor cuando está en conducción. Anotar los resultados en la tabla.

	Diodo led	Diodo semiconductor
Voltaje en conducción		

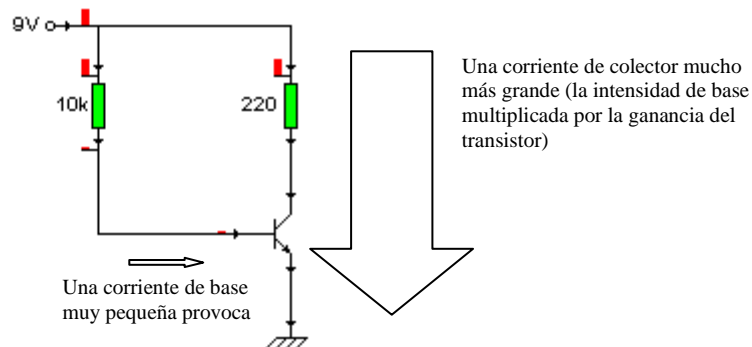
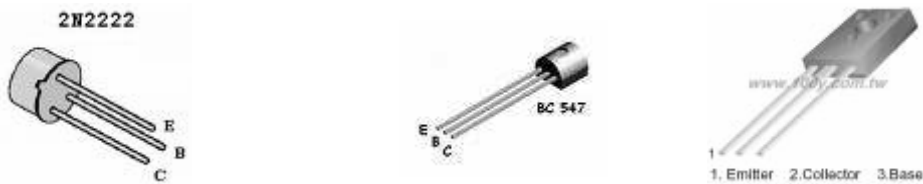
TRANSISTORES

Un **transistor** es un dispositivo electrónico que tiene tres terminales denominados emisor, base y colector. Su principal característica es que se trata de un dispositivo “amplificador”. Si, en las condiciones adecuadas, introducimos una pequeña corriente eléctrica en la base, ésta se amplifica (se hace mucho mayor) y circula entre el colector y el emisor. El factor por el que se amplifica la corriente de base se llama ganancia de corriente del transistor, β (su valor puede variar de 40 a 400, dependiendo del tipo de transistor). En definitiva, podemos escribir que $I_c = \beta \cdot I_b$



Cualquier pequeña variación en la corriente de base se verá considerablemente amplificada en el colector. El transistor emplea una corriente pequeña (la de base) para conectar o modificar una corriente más grande (la de colector). Lógicamente, si no hay corriente en la base tampoco circulará corriente entre el colector y el emisor.

En las figuras adjuntas vemos varios tipos de transistores y la disposición de sus tres patillas. Ésta hay que tenerla muy en cuenta, para no confundir las patillas a la hora de montarlo en un circuito.

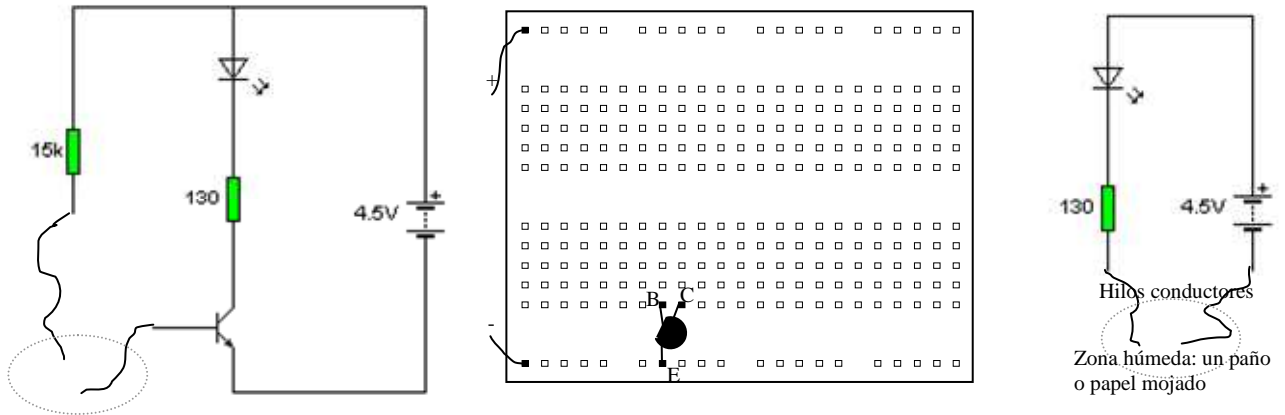


Al igual que los diodos, los transistores se fabrican con silicio. En este caso, se coloca una zona P en medio de dos zonas N (transistor NPN), o bien, una zona N entre dos P (transistor PNP). Los transistores que aparecen en las figuras anteriores son NPN. En los PNP los voltajes y las corrientes tienen sentidos contrarios y en el símbolo la flecha del Terminal de emisor apunta hacia dentro.

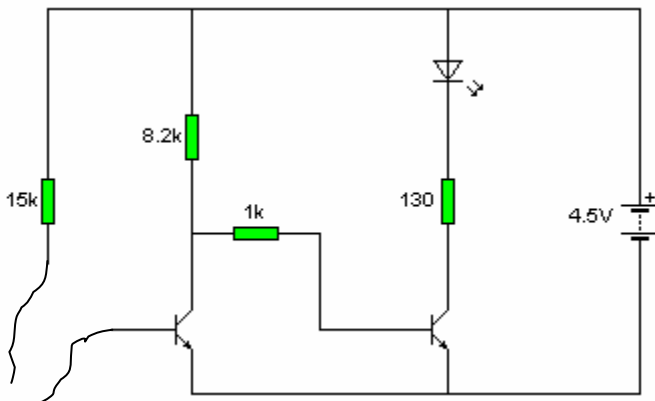
ACTIVIDAD 4.- EL TRANSISTOR. DETECTOR DE HUMEDAD.

El agua (no destilada) es un buen conductor de la electricidad, aprovechando esa propiedad pretendemos construir un circuito que haga encenderse un diodo led cuando unos hilos de cobre, que hacen de sondas, estén en una zona húmeda.

4.1.- Construir el circuito adjunto y comprobar que el led se ilumina cuando tocamos la zona húmeda con los hilos. Repasar el comportamiento de un transistor y explicar por que se enciende el diodo.



4.2.- En el circuito de la figura el diodo led se enciende cuando los hilos están secos, manteniéndose apagado cuando los hilos se encuentran en una zona húmeda. Podría servir, por ejemplo, para indicarnos si hay que regar una maceta. Montar el circuito y explicar su funcionamiento, indicando previamente en la tabla adjunta el estado (conduce-no conduce) de los transistores y del diodo led.

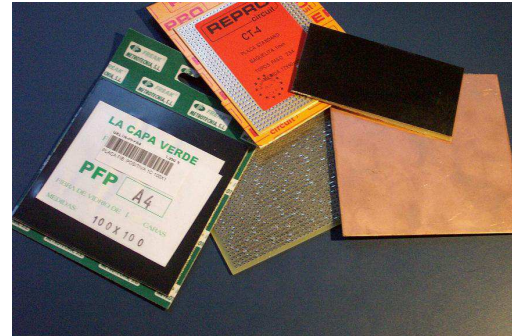


	Primer transistor	Segundo transistor	Diodo led
Sondas húmedas			
Sondas secas			

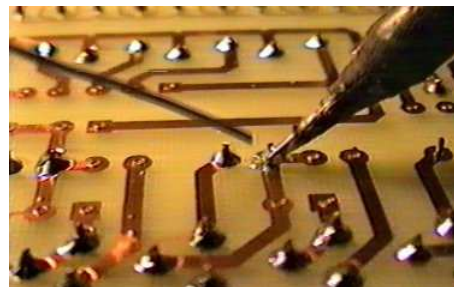
CONSTRUCCIÓN DE CIRCUITOS IMPRESOS.

Los circuitos electrónicos se montan sobre unas placas de baquelita o fibra de vidrio (materiales aislantes) que tienen en su parte inferior una capa de cobre (conductor) que sirve de base para realizar las uniones entre los distintos componentes, los cuales se soldarán finalmente a la misma.

Para realizar las conexiones sobre la capa de cobre tenemos que utilizar un rotulador resistente a los ácidos, con el que dibujaremos las pistas que unen los componentes siguiendo el esquema correspondiente. Posteriormente se introduce la placa en una solución ácida que hará desaparecer todo el cobre sobre el que no de haya dibujado, quedando solamente las conexiones del circuito.



Los componentes colocados en la parte superior



La parte inferior ya atacada por el ácido, en la que queda el cobre correspondiente a las pistas dibujadas

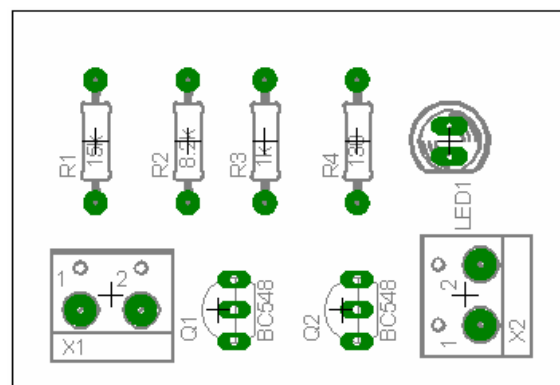
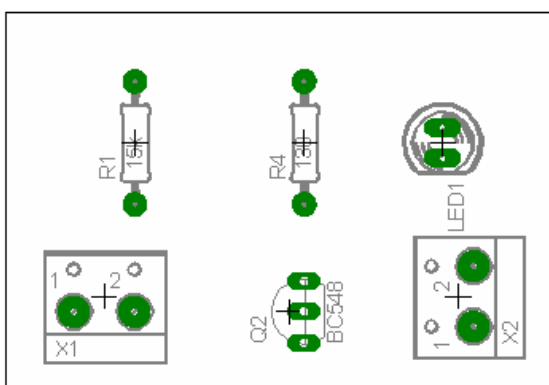
El proceso a seguir para diseñar una placa de circuito impreso sería, pues, el siguiente:

1. Distribuir los componentes, marcando a tamaño real los puntos correspondientes a sus patillas, sobre una hoja de papel. En estos puntos habrá que dibujar una isla redonda y, una vez realizado el circuito taladrar para insertar los componentes.
2. Dibujar las conexiones entre los distintos elementos siguiendo el esquema correspondiente. Tener en cuenta que ahora no se pueden cruzar pistas entre las que no haya conexión eléctrica.
3. Copiar el dibujo en la parte trasera del papel, este será el que tenemos que transferir al cobre.
4. Volver a dibujar el circuito en la placa de cobre con un rotulador permanente (resistente a los ácidos).
5. Introducir la placa en una solución especial que eliminará el cobre que no esté cubierto por el rotulador.

La solución que utilizamos para atacar el cobre la podemos hacer de la siguiente manera:

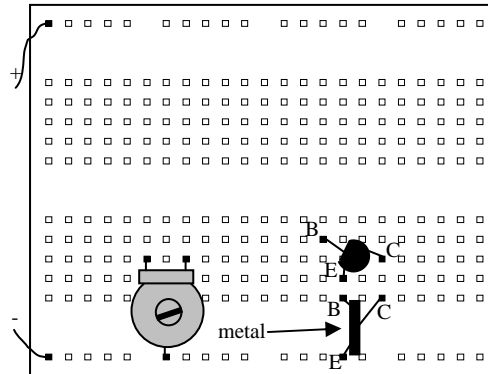
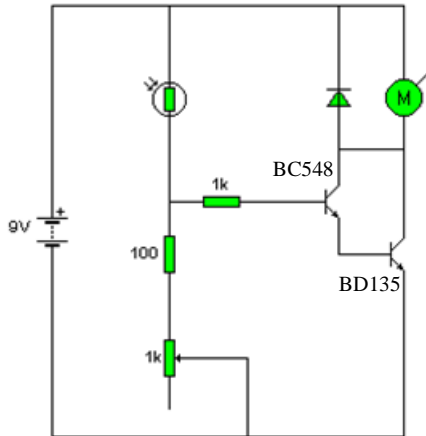
1. Una o dos partes de agua.
2. Una parte de agua fuerte (ácido clorhídrico).
3. Una parte de agua oxigenada de 110 volúmenes.

ACTIVIDAD 5.- Dibujar sobre las figuras adjuntas las pistas de cobre que unirían los distintos componentes correspondientes a los circuitos de los ejercicios 4.1 y 4.2



ACTIVIDAD 6.- DETECTOR DE LUZ.

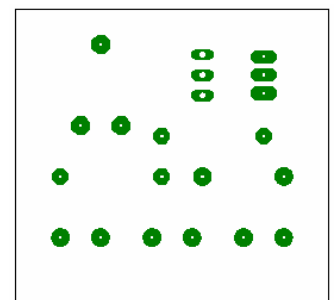
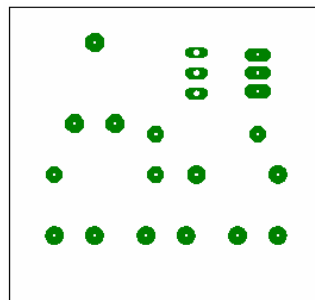
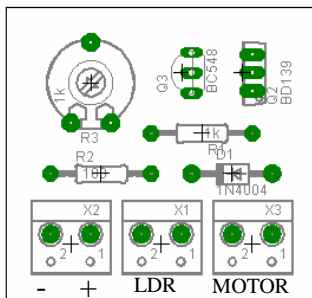
Con el circuito de la figura adjunta pretendemos conseguir que el motor gire cuando incida luz sobre la LDR. Cuando esto es así, la resistencia de la LDR es baja y puede pasar suficiente corriente a la base del primer transistor para que, una vez amplificada, provoque el movimiento del motor. Cuando no incide luz sobre la LDR su resistencia es mucho más alta, impidiendo que los transistores conduzcan y, por tanto, que gire el motor.



En primer lugar, montaremos el circuito en una placa de pruebas para comprobar su funcionamiento y, posteriormente, tenemos que diseñar y construir el circuito impreso. Finalmente colocaremos y soldaremos los componentes.

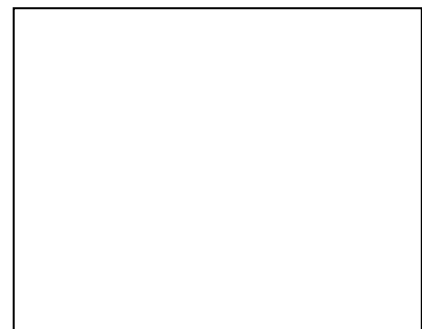
6.1 En las figuras inferiores, dibujar las pistas que deben unir las patillas de los componentes, teniendo en cuenta el esquema (recuerden que estas pistas no se pueden cruzar).

Circuito visto desde el lado de los componentes, las zonas para las soldaduras estarían en la cara posterior (la del cobre)



6.2.- Volver a realizar el dibujo en un folio en blanco, dar la vuelta a la hoja y copiar el dibujo definitivo por esa cara. Estas serán las pistas que, definitivamente, hay que dibujar con un rotulador permanente sobre la placa de cobre (previamente cortada al tamaño adecuado). Repetir el diseño en el cuadro adjunto.

6.3.- Una vez seco el rotulador, introducir la placa en la solución atacadora hasta que desaparezca el cobre que no está protegido por la tinta del rotulador (tomar las precauciones de seguridad necesarias).



6.4.- Colocar los componentes y soldarlos.

CONDENSADORES

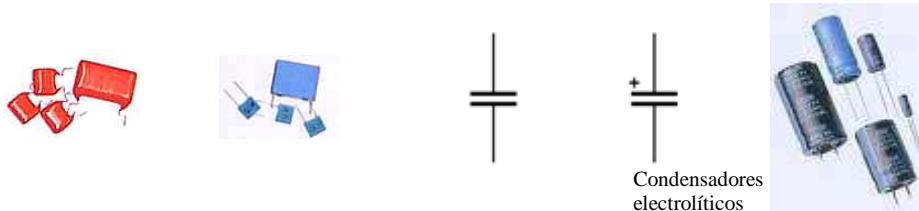
Los **condensadores** están formados por dos placas metálicas separadas por un material aislante. Tienen la propiedad de almacenar electrones (carga negativa) en una de sus placas y tener falta de ellos (carga positiva) en la otra. Esta “**capacidad**” se mide en **faradios (F)** y sus divisores μF , nF y pF

$$1\mu\text{F} = 10^{-6} \text{ F}$$

$$1\text{nF} = 10^{-9} \text{ F}$$

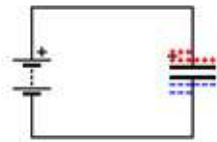
$$1\text{pF} = 10^{-12} \text{ F}$$

Algunos condensadores tienen una polaridad fija, es decir una de sus placas debe ser siempre la que se cargue positivamente y la otra negativa, son los denominados condensadores electrolíticos. En la figuras adjuntas podemos ver varios tipos de condensadores y su símbolo:



Puesto que los condensadores se cargan almacenando electrones en una de sus placas, posteriormente, pueden descargarse y suministrar los electrones almacenados al circuito en el que estén conectados.

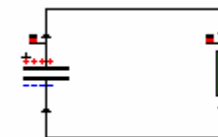
Al conectar un condensador a los extremos de una pila sus placas adquieren la misma diferencia de cargas existente entre los bornes de la pila



Si, después, quitamos el condensador este permanecerá cargado



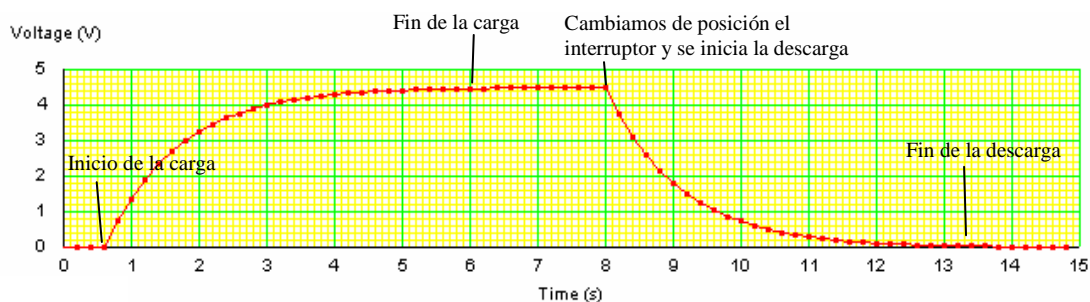
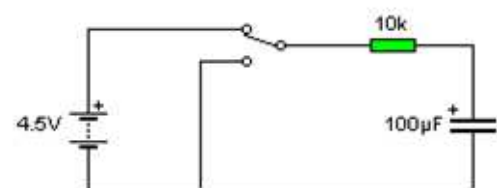
Y si lo conectamos a algún elemento resistivo, será capaz de suministrar una corriente eléctrica, que durará el tiempo que tarde en descargarse completamente



Hay que tener en cuenta que un condensador se carga, pero no tiene un mecanismo interno que permita mantener la carga separada (como hacen las pilas), por eso, el condensador puede suministrar corriente eléctrica durante un pequeño intervalo de tiempo (el que dura su descarga), pero no de manera permanente. En la figura anterior la carga del condensador se producía de manera instantánea, ya que entre el condensador y la pila no había ninguna resistencia; lógicamente, de haberla, la carga se iría produciendo poco a poco. El tiempo que dura la carga o la descarga de un condensador, a través de una resistencia, dependerá del valor de la resistencia presente en el circuito y de la capacidad del propio condensador. Aproximadamente se puede calcular multiplicando 5 por la capacidad del condensador (expresada en faradios) y por la resistencia (expresada en ohmios).

$$T = 5 \cdot R \cdot C$$

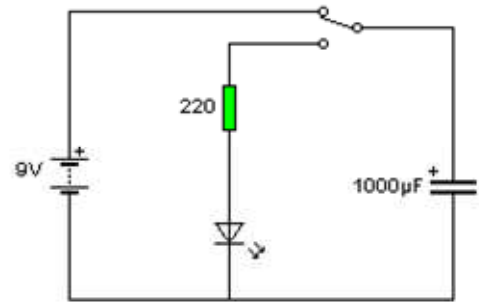
En el circuito de la figura se puede estudiar el proceso de carga y descarga del condensador. Cuando ponemos el interruptor en el contacto superior el condensador comienza su carga, a través de la resistencia, y el voltaje entre sus extremos va aumentando progresivamente. Una vez cargado, si movemos el interruptor al contacto inferior se iniciará la descarga. En la gráfica inferior vemos la evolución del voltaje entre los extremos del condensador a lo largo de todo el proceso:



ACTIVIDAD 7.- SIMULACIÓN DE LA CARGA Y DESCARGA DE UN CONDENSADOR.

7.1.- Utilizando el programa Cocodrile clips realizar el circuito de la figura.

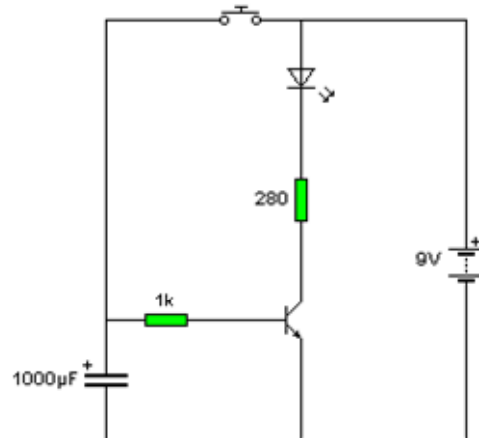
7.2.- Indicar la posición que tiene que tener el interruptor (superior o inferior) para que el condensador se cargue y se descargue.



7.3.- ¿Se enciende el led cuando el condensador se descarga? Si la respuesta es afirmativa indicar cuanto (mucho, regular, poco, muy poco).

7.4.- Cambiar la resistencia por una de 10K. ¿Se enciende el led cuando el condensador se descarga?. Explicar la respuesta.

7.5.- Montar el circuito adjunto. Explicar que procesos tienen lugar cuando se acciona el pulsador y cuando se suelta.



7.6.- Medir el tiempo que permanece encendido el diodo led para los valores de resistencia de base indicados.

Resistencia de base	Tiempo encendido led
1kΩ	
10KΩ	
100KΩ	

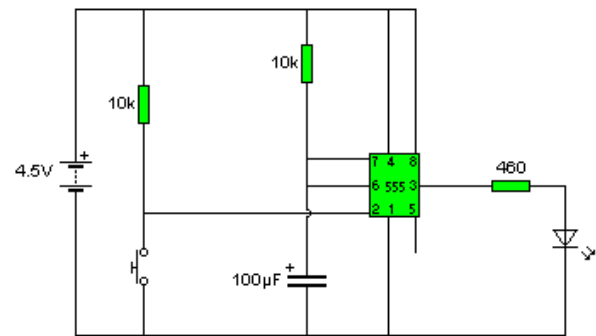
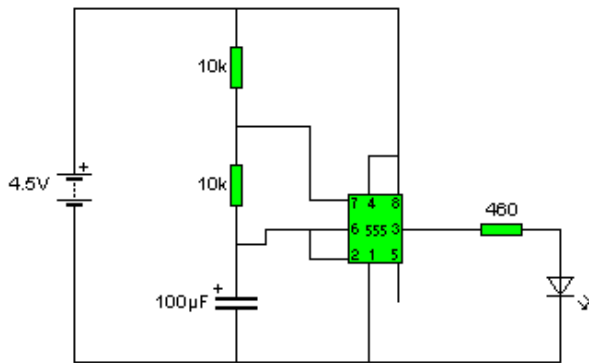
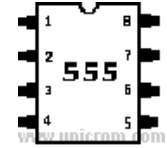
7.7.- ¿Qué efecto tiene el transistor sobre el encendido del diodo led (comparar con lo que sucedía en el primer circuito)?

EL CIRCUITO INTEGRADO 555

Los circuitos integrados constan de una pequeña placa de silicio (chips) instalada en una cajita plástica y conectada al exterior por sendos juegos de patillas situados en los laterales. Sobre la placa de silicio se diseñan circuitos “microscópicos” formados por muchos transistores (desde unas decenas hasta millones) capaces de dar solución a las más diversas aplicaciones.



El circuito integrado 555 sirve, principalmente, para controlar el tiempo de encendido o apagado de cualquier dispositivo conectado a su salida, para ello toma como base la carga y descarga de un condensador a través de una resistencia. En las figuras siguientes se muestran dos de sus aplicaciones, el circuito de la izquierda hace parpadear un diodo led y en el de la derecha enciende un diodo led por un tiempo determinado cuando accionamos el pulsador.



ACTIVIDAD 8.- SIMULACIÓN DE TEMPORIZADORES CON EL CI 555

8.1.- Montar el primer circuito utilizando el programa Cocodrilo clips. Comprobar y explicar su funcionamiento.

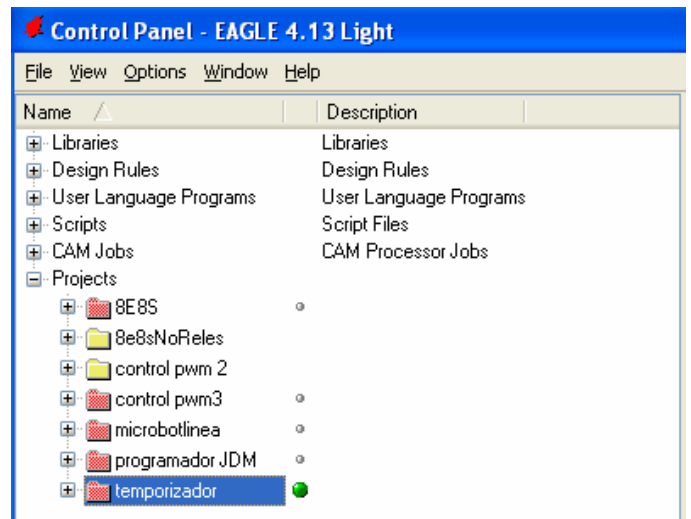
8.2.- ¿Qué componentes determinan la velocidad a la que el led parpadea?. Modificarlos, comprobar los resultados y anotar los valores y los tiempos correspondientes (para tres ejemplos distintos).

8.3.- Montar el segundo circuito y comprobar y explicar su funcionamiento.

8.4.- ¿Qué componentes determinan el tiempo de encendido del diodo led?. Modificarlos, comprobar los resultados y anotar los valores y los tiempos correspondientes (para tres ejemplos distintos).

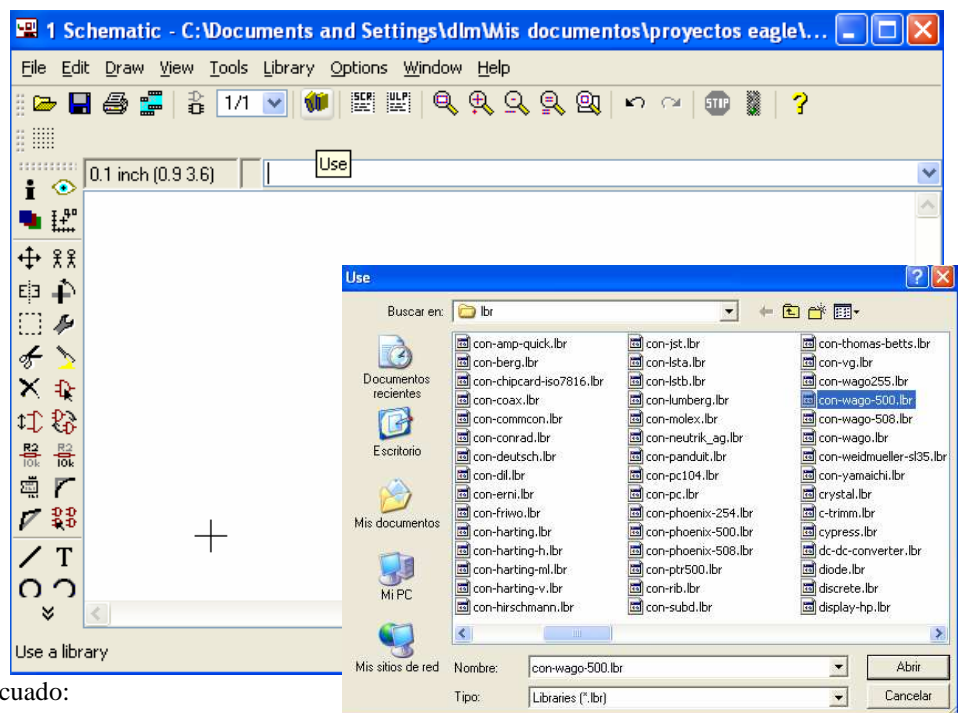
ACTIVIDAD 9.- CIRCUITO IMPRESO POR ORDENADOR. EAGLE es un potente editor de gráficos y esquemas para el diseño de placas de circuito impreso con el ordenador.

Al iniciar el programa aparece el panel recontrol de la figura. Para iniciar un proyecto tenemos que seleccionar File---New---Project, de esta manera se crea una carpeta para guardar el proyecto a la que podremos darle el nombre que deseemos (temporizador). Ahora, teniendo activada la carpeta seleccionamos File---New---Schematic para dibujar el esquema del circuito.



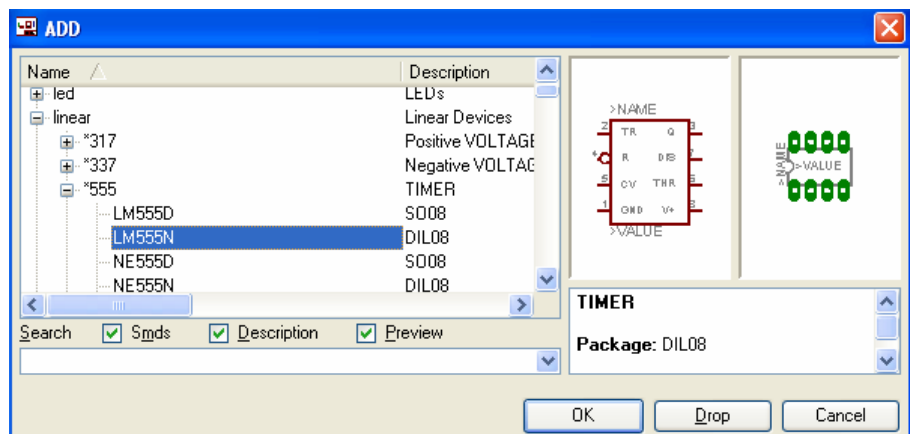
Para introducir los componentes en el esquema tenemos, previamente, que cargar las librerías oportunas. Esto se consigue pinchando en el icono Use (Library---Use) y vamos seleccionando las bibliotecas adecuadas. En nuestro trabajo tenemos que abrir las siguientes librerías:

- con-wago-500.lbr
- diode.lbr
- led.lbr
- linear.lbr.
- pot.lbr
- rcl.lbr
- supply1.lbr
- transistor.lbr



Ahora podemos añadir símbolos al esquema pinchando en el icono Add (Edit---Add) y seleccionando el componente adecuado:

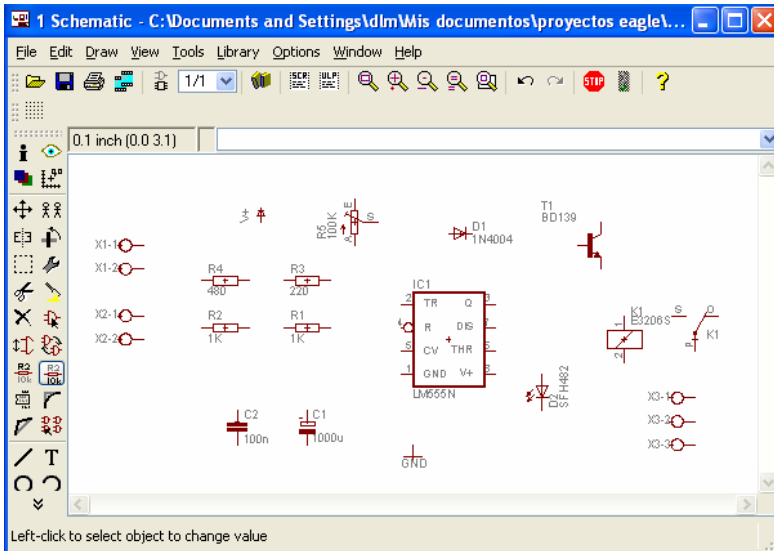
- LM555N
- CPOL-EUE5-13
- C-EU102-043X133
- R-EU_0207/12
- TRIM_EU-LI15
- 1N4004
- SFH482
- E3206S
- BD139
- W237-103
- W237-102
- V+
- GND



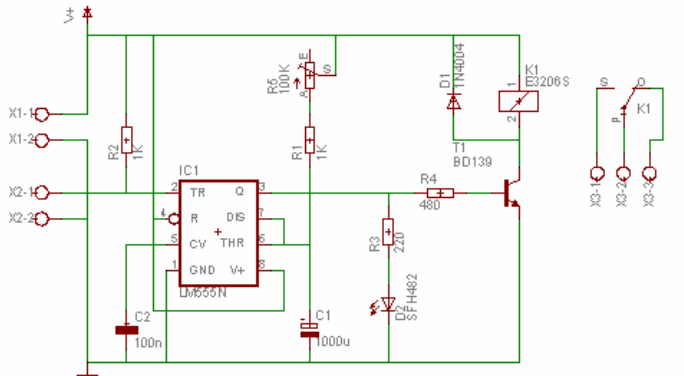
Utilizando el icono Value (Edit---Value) podemos dar los valores correspondientes a las resistencias y los condensadores.

Finalmente construimos nuestro esquema moviendo los componentes a los lugares adecuados (Move, Rotate, etc.) y dibujando las líneas de conexión (Wire).

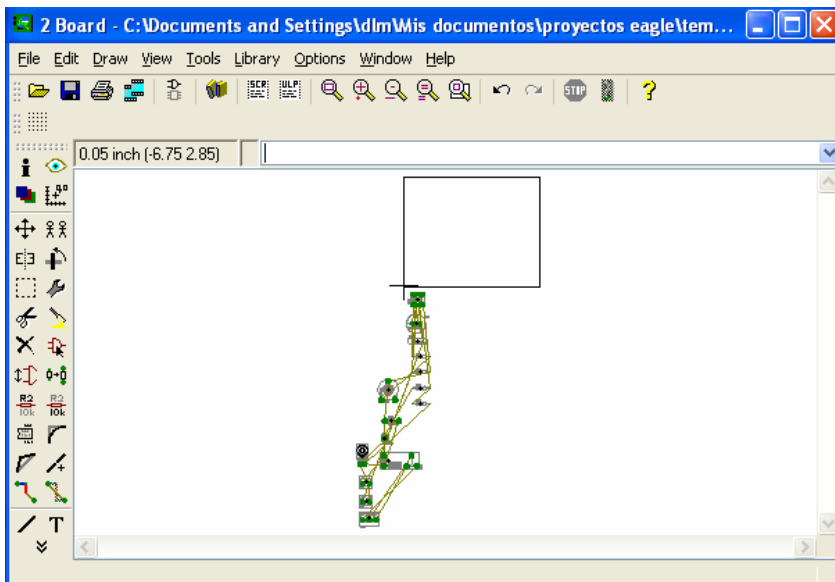
Para realizar una acción (mover, rotar, etc.) sobre un grupo, lo seleccionamos con el icono Group y después pinchamos con el botón derecho del ratón.



El esquema completo es el de la figura adjunta.

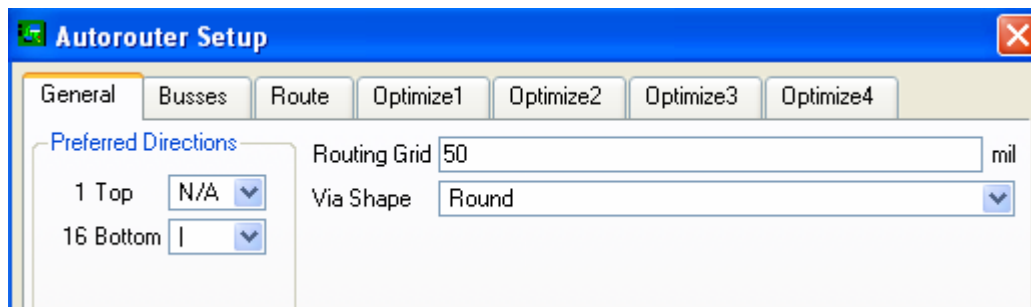


Pinchando en el icono Board pasamos a la pantalla que nos permite hacer el diseño del circuito impreso.



En primer lugar tenemos que colocar los componentes dentro del área de la placa del c.i. Esta es una operación importante, ya que según coloquemos los componentes el dibujo posterior de las pistas será más o menos complicado de realizar (en principio, tendríamos que procurar que los cruces de las conexiones sean los menores posibles)

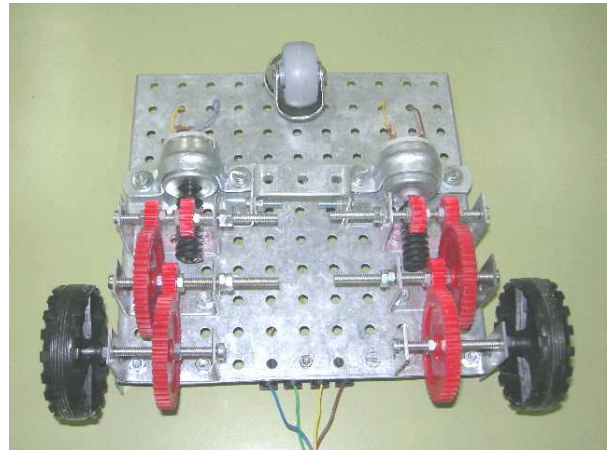
Una vez colocados los componentes podemos dibujar las pistas manualmente (pinchando en el icono Route) o dejar que lo haga el programa de manera automática (pinchando en el icono Autoroute).



ACTIVIDAD 10.- CONSTRUIR UN ROBOT QUE BUSCA LA LUZ.

En la fotografía adjunta se muestra la parte mecánica del robot (boca abajo). Lleva dos ruedas traseras, que se mueven de manera independiente a través de dos motores (con su sistema de reducción de velocidad), y una rueda loca delantera. De esta manera, si los dos motores giran hacia delante el móvil irá en ese sentido en línea recta, pero si uno de los motores se para el vehículo gira hacia el lado del motor detenido.

Nuestro vehículo que busca la luz arrancará inicialmente con un de los motores parado, de esta manera girará sobre sí mismo, debido a la acción del motor que está girando. Cuando detecte un haz de luz se pondrá, también, en marcha el motor parado y el móvil se dirigirá hacia la luz en línea recta.



10.1.- Construir un vehículo similar al de la fotografía, añadiéndole el detector de luz que se ha construido en la actividad 5.

10.2.- Calcular la relación de transmisión de los sistemas de reducción de velocidad de los motores.

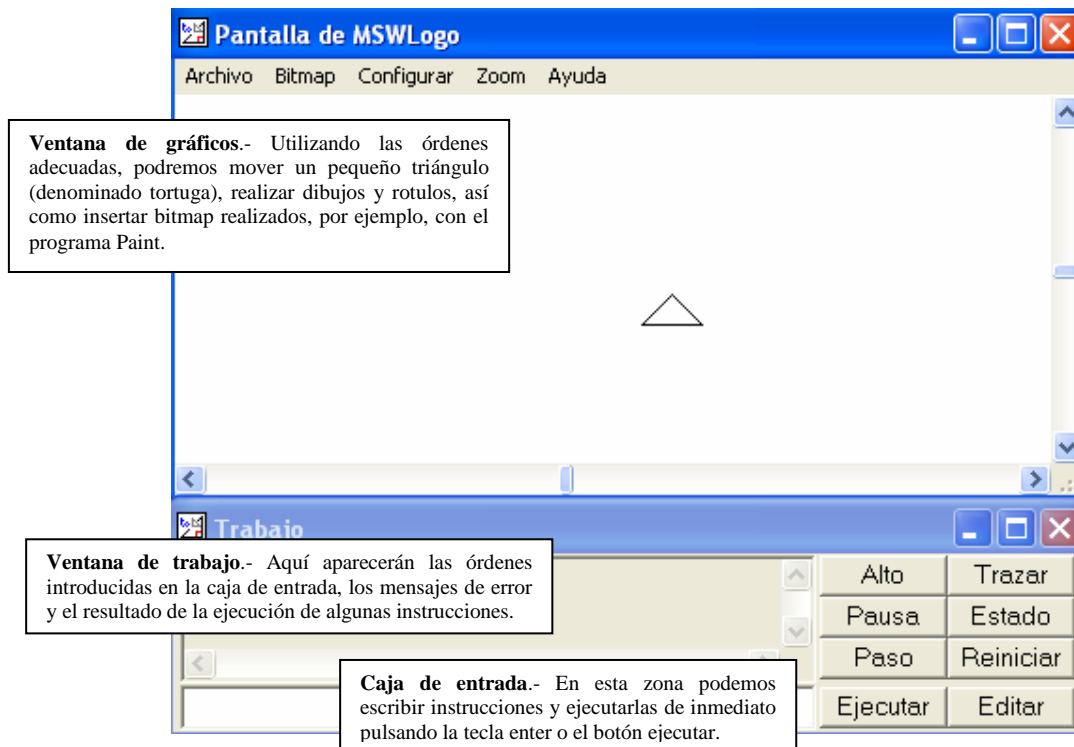
10.3.- Dibujar el esquema completo del robot.

10.4.- Realizar una lista con el nombre y la cantidad de todos los materiales utilizados, tanto mecánicos como electrónicos.

MSWLOGO: 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

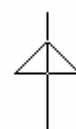
Ejemplo: av 100



retrocede distancia

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

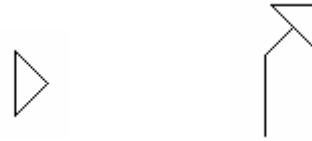
Ejemplo: av 100 re 50



giraderecha ángulo

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

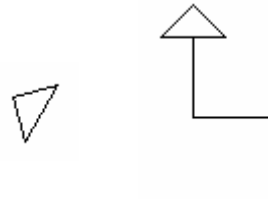
Ejemplo: gd 90
Ejemplo: av 40 gd 45 av 20



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.

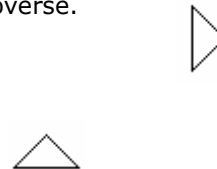
Ejemplo: gi 60
Ejemplo: av 40 gi 90 av 40 gd 90 av 40



subelapiz

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

Ejemplo: sl av 50 gd 90 av 50



bajalapiz

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

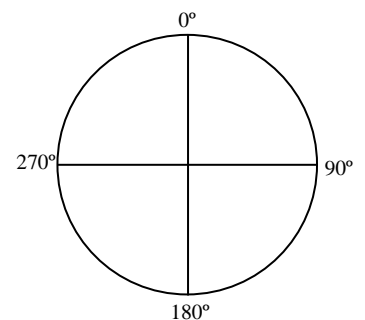
Ejemplo: gd 90 av 50 sl av 50 bl av 50



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.

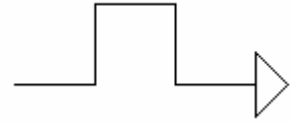
Ejemplo: ponrumbo 90



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.

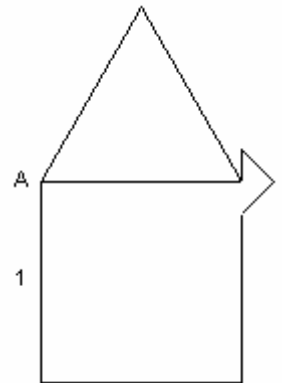
3. Introducir las instrucciones necesarias (utilizar av, gi y gd) para realizar el dibujo adjunto. Las líneas son de 40 puntos.



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

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

5. 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.



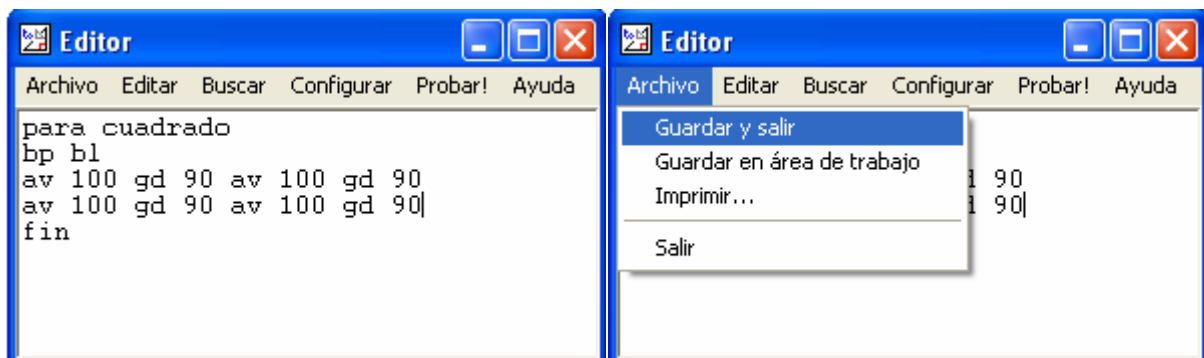
PROCEDIMIENTOS

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.

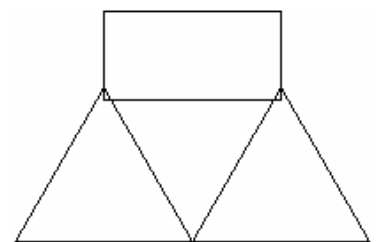


Después de escribir un procedimiento y guardarlo en la memoria del ordenador, éste puede ser tratado como si hubiésemos creado una nueva instrucción y, como tal, lo podemos ejecutar incluyéndolo en otros procedimientos. En general, los programas se suelen dividir en varios procedimientos que son llamados o ejecutados desde uno principal o inicial.

ACTIVIDADES:

- 6. Realizar un procedimiento denominado `rectángulo` y otro llamado `triángulo` que dibujen un rectángulo y un triángulo similares a los de las actividades 1 y 2 anteriores.

- 7. Escribir un procedimiento denominado `figura` que realice el dibujo adjunto. Los triángulos y el rectángulo se harán llamando a los procedimientos creados en el ejercicio anterior e intercalando las instrucciones necesarias para orientar correctamente la tortuga y colocarla en la posición adecuada.



```
para rectangulo           para figura
```

```
fin
```

```
para triangulo
```

```
fin
```

```
fin
```

LA INSTRUCCIÓN REPITE.

Como su nombre indica sirve para repetir una o varias instrucciones un número determinado de veces.

repite nº de veces [instrucciones a repetir]

Repite el nº de veces indicado las instrucciones encerradas entre corchetes.

Ejemplo: Para dibujar un cuadrado podemos repetir 4 veces las instrucciones `av 100 gd 90`. Es decir:
`repite 4 [av 100 gd 90]`

ACTIVIDADES:

- 8. Comprobar el ejemplo anterior y utilizar la instrucción `repite` para dibujar un rectángulo de 100 x 50.
- 9. Realizar un procedimiento (lo llamaremos `apilar`) que dibuje 4 cuadrados de lado 20 uno encima del otro, pero separados una distancia de 5 puntos (utilizar la instrucción `repite`).

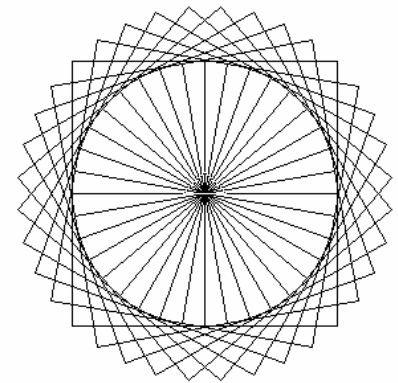
```
para cuadrado           para apilar
                        repite 4 [cuadrado
                        fin
fin
```



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

```

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

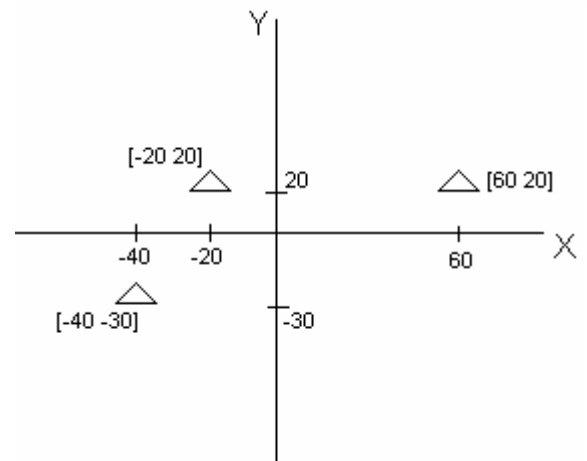


LAS COORDENADAS DE LA TORTUGA. PONPOS.

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:



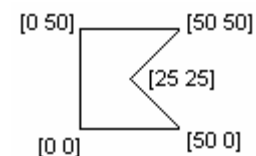
ponpos [coordenadaX coordenadaY]

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

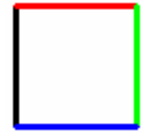
Esta instrucción nos indica el color del punto de la pantalla (píxel) en el que se encuentra la tortuga, mostrándonos los números que representan la intensidad de los colores rojo, verde y azul correspondientes.

pon grosor [altura anchura]

pon g [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
pon g [3 3]
pon cl [0 0 0] av 60 gd 90
pon cl [255 0 0] av 60 gd 90
pon cl [0 255 0] av 60 gd 90
pon cl [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 pon color relleno y rellena:

pon color relleno [rojo verde azul]

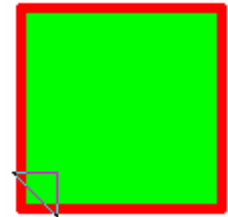
pccr [rojo verde azul] Pone el color de relleno indicado

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.

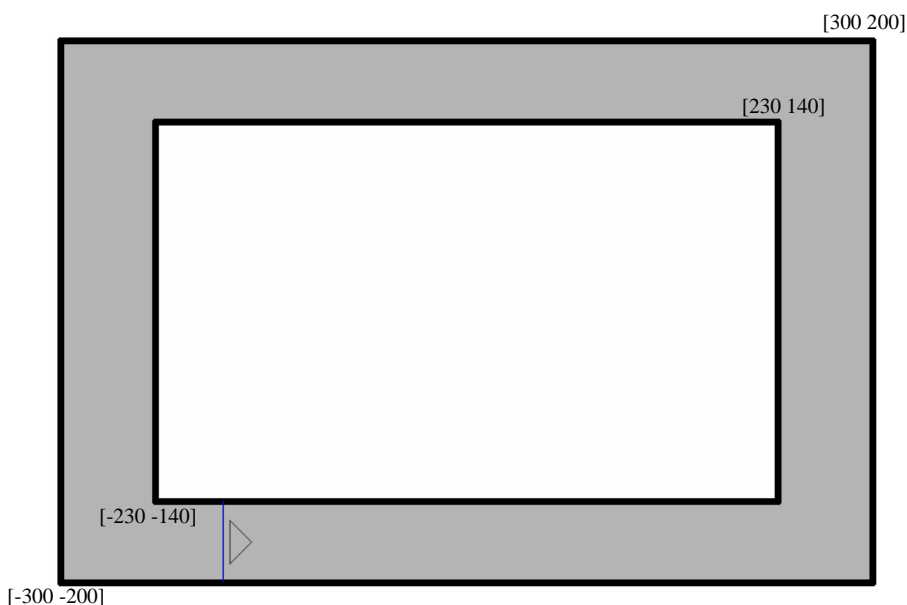
Ejemplo: Vamos a dibujar un cuadrado de color rojo relleno de verde:

```
para cuadrado de color
pon color lapiz [255 0 0] pon grosor [5 5]
repite 4[av 100 gd 90]
sl gd 45 av 10 bl
pon color relleno [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 20º a la derecha

ACTIVIDADES:

14. Comprobar el procedimiento denominado juego que contiene las instrucciones ponfoco y ponteclado, de manera que la tortuga girará 20º 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.

caracter código ASCII

car código ASCII Devuelve el carácter (letra, número, signo,...) correspondiente al código ASCII indicado.

Estas dos instrucciones las usamos de manera combinada, es decir:

car lc Así, obtenemos el carácter correspondiente a la última tecla pulsada.

La letra, número o símbolo obtenido con las instrucciones anteriores tenemos que almacenarla en algún lugar, para ello utilizamos las variables:

Una variable es una porción de la memoria del ordenador donde podemos guardar algo: un número, una palabra, un carácter, una frase, etc. A las variables hay que asignarles un nombre y un valor inicial (su contenido, que posteriormente podrá ser modificado). Para crear una variable se utiliza la instrucción haz:

haz "nombre de la variable valor asignado"

Ejemplo: `haz "operando 5` define la variable operando y le asigna un valor inicial de 5
 `haz "tecla car lc` define la variable tecla y, a través de car lc, le asigna el carácter de la tecla pulsada, que queda almacenado en la variable.

Para referirnos, posteriormente, al valor de una variable, es decir, a lo que guarda en su interior, tenemos que colocar dos puntos (:) inmediatamente delante de su nombre.

TOMANDO DECISIONES. SI

En la mayoría de los programas el ordenador necesita tomar decisiones; es decir, decidir la realización de unas actuaciones u otras en función del resultado de una operación, de que un número sea mayor o menor que otro, de que una condición sea verdadera o falsa, etc.

si condición [instrucciones]

Si la condición es verdadera se ejecutan las instrucciones comprendidas entre los corchetes. Si la condición es falsa se pasa a la siguiente instrucción del programa sin ejecutar las encerradas entre los corchetes. Las condiciones pueden ser de distintos tipos: =, >, <, etc.

Ejemplo: El siguiente programa activa los eventos del teclado con ponfoco y ponteclado y cada vez que se pulsa una tecla se ejecuta el procedimiento control. En él se crea la variable "tecla que guardará el carácter de la última tecla pulsada, para preguntar después si esa tecla es la r o la c y, así, dibujar un rectángulo o un cuadrado.

<code>para figuras</code>	<code>para control</code>
<code>ponfoco [Pantalla de MSWLogo]</code>	<code>haz "tecla car lc</code>
<code>ponteclado [control]</code>	<code>si :tecla="r" [rectangulo]</code>
<code>fin</code>	<code>si :tecla="c" [cuadrado]</code>
	<code>fin</code>

ACTIVIDADES:

16. Comprobar el ejemplo anterior añadiendo los procedimientos rectangulo y cuadrado correspondientes.

17. Modificar el procedimiento juego del ejercicio 14 para que la tortuga gire a la izquierda 20° cada vez que pulsamos la tecla "4" y 20° a la derecha si pulsamos la tecla "6".

<code>para juego</code>	<code>para control</code>
<code>ponfoco [Pantalla de MSWLogo]</code>	
<code>ponteclado [control]</code>	
<code>fin</code>	<code>fin</code>

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 volverá 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.

INTRODUCIR DIBUJOS CON PAINT. CARGADIB.

Para llevar un dibujo realizado con Paint a la pantalla de MSWLogo utilizamos la instrucción cargadib:

cargadib nombrebitmap

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. Nombrebitmap 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. Los bordes deben ser negros y la línea de salida de color azul. Tener en cuenta que habrá que utilizar ponpos y ponrumbos para que, una vez cargado el circuito, coloquemos la tortuga delante de la línea de salida.

CEROS Y UNOS.

Los ordenadores son dispositivos que internamente trabajan usando el denominado sistema binario de numeración. En él sólo se dispone de dos dígitos, el "cero" y el "uno".

0 "cero" significa ausencia de corriente eléctrica, desactivado, desconectado,...
1 "uno" significa presencia de corriente eléctrica, activado, conectado,...

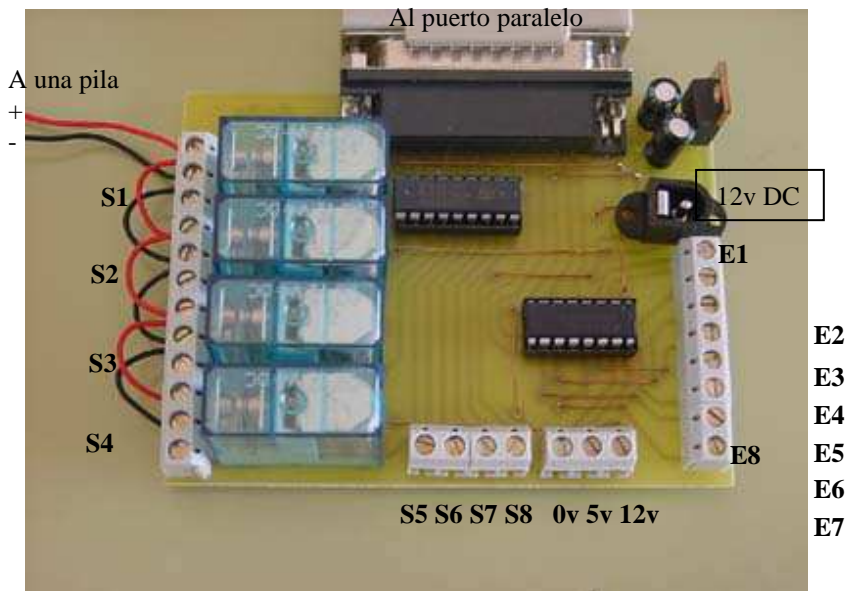
Ellos entienden estos dos dígitos y pueden suministrarlos para activar y desactivar elementos externos (motores, bombillas, timbres, electroimanes,...). Del mismo modo, también aceptarían órdenes externas dadas en dicho formato, mediante interruptores o pulsadores que permitan o no el paso de la corriente eléctrica.

PLACA CONTROLADORA.

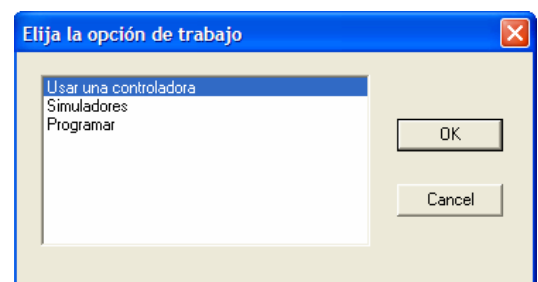
Para conectar el ordenador con el mundo exterior es necesario una placa controladora que proporcione las salidas y las entradas oportunas. En nuestro caso disponemos de una placa controladora que

dispone de 8 salidas y 8 entradas, en ambos casos digitales; es decir, trabajan exclusivamente con "ceros" (tensión eléctrica próxima a 0v) y "unos" (tensión eléctrica de 5v). La unión con el ordenador se realiza a través del puerto paralelo. La placa se alimenta mediante una fuente externa de 12v.

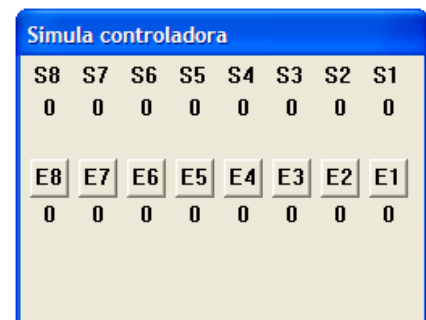
Las salidas S1, S2, S3 y S4 están formadas por sendos relés, tal como se aprecia en la figura anterior. S5, S6, S7 y S8 se corresponden con salidas a transistor en colector abierto.



Para utilizarla seguiremos necesitando MSWLogo. Con él realizaremos los programas que hagan funcionar los dispositivos que conectemos a las salidas y entradas de la placa. Para ello, cuando se inicia MSWLogo y aparece la ventana "Elija una opción", tenemos que seleccionar "Usar una controladora". De esta manera se cargarán unos procedimientos especiales (salidas y leerentradas) que nos permitirán comunicarnos con la placa.



Normalmente, antes de probar un programa directamente en la controladora, lo simularemos utilizando sim8e8s (se elige la opción simuladores y se selecciona sim8e8s de la lista que aparece). Nos aparecerá en pantalla una ventana que responderá al programa que realicemos poniendo a uno o a cero las salidas oportunas y simulando las entradas mediante botones que podemos pinchar con el ratón.



INSTRUCCIONES DE CONTROL.

salidas :n

Activa (pone a 1) las salidas correspondientes al número indicado, según la tabla adjunta. Se pueden conectar varias salidas a la vez sumando los números correspondientes a cada una de ellas.

S8	S7	S6	S5	S4	S3	S2	S1
128	64	32	16	8	4	2	1

Ejemplo:

```
salidas 0   Desconecta todas las salidas.
salidas 8   Activa la salida 4, desconecta todas las demás.
salidas 3   Activa las salidas 1 y 2, desconecta todas las demás.
salidas 38  Activa las salidas 6, 3 y 2, desconecta todas las demás.
```

Al ejecutar esta instrucción se actualizan a 0 o a 1 unas variables correspondientes a cada una de las salidas ("s1, "s2,..., "s8) y, por tanto, si es necesario, podremos preguntar por su estado desde cualquier lugar de nuestros programas.

Ejemplo: si :s1=1 [salidas 0] ;si s1 está en 1 desactiva todas las salidas
 si :s1=0 [salidas 1] ;si s1 está en 0 activa la salida 1 y desactiva las demás

También disponemos de la variable "sn que guarda, en todo momento, el número decimal correspondiente al estado de las ocho salidas.

Ejemplo: si :s3=0 [salidas :sn+4] ;activa la salida 3 sin modificar el resto

 si :s3=1 [salidas :sn-4] ;desactiva la salida 3 sin modificar el resto

leerentradas

Realiza un chequeo del valor (0 o 1) que hay en cada una de las entradas, almacenándolo en las variables: "e1, "e2, "e3, ..., "e8. Para saber, posteriormente, en que estado se encuentra una determinada entrada tenemos que preguntar por el valor de la variable correspondiente.

Ejemplo: leerentradas
 si :e1 = 1 [salidas 1] ;si la entrada 1 vale 1 (5v) activa la salida 1

Al igual que con las salidas, también se crea la variable "en para almacenar el número decimal correspondiente al estado de todas las entradas.

EL CONTROL DEL TIEMPO. ESPERA.

En los programas de control es muy frecuente que tengamos la necesidad de realizar temporizaciones, es decir, controlar el tiempo que un dispositivo está encendido o apagado. MSWLogo cuenta con una instrucción específica para este trabajo, se trata de espera:

espera tiempo

Detiene la ejecución de un procedimiento durante el tiempo especificado (éste se expresa en 1/60 segundos), una vez transcurrido continúa con la ejecución del resto de instrucciones.

Ejemplos: espera 1 Detiene el programa durante 1/60 segundos (0'0166 segundos).
 espera 60 Detiene el programa durante 1 segundo.
 espera 300 Detiene el programa durante 5 segundos.
 espera 3600 Detiene el programa durante 1 minuto.

ACTIVIDADES: (utilizar sim8e8s para comprobar las actividades)

1. Indicar que hace el siguiente programa:

```
para actividad1
salidas 3 espera 600
salidas 2 espera 300
salidas 0
fin
```

2. Indicar que hace el siguiente programa:

```
para actividad3
ponfoco [Pantalla de MSWLogo]
ponteclado [control]
fin
```

```
para control
haz "tecla car lc
si :tecla = "a [salidas 1 espera 480 salidas 0]
fin
```

3. Control marcha-paro de un motor. Se dispone de un motor conectado a la salida 1 y se quiere controlar con la teclas m (marcha) y p (parada). Programa el control de dicho motor.

4. Un motor, conectado a las salidas S1 y S2, debe girar a derecha durante 5 segundos, posteriormente parar durante 2 seg. Y finalmente girar a izquierda durante 10 seg. antes de detenerse. Realizar el programa adecuado, teniendo en cuenta que el inicio del ciclo tiene lugar accionando un pulsador conectado a la entrada E1.

5. Tenemos 8 bombillas conectadas a las 8 salidas de la placa controladora y queremos que se enciendan progresivamente, de una en una, a intervalos de 0'5 s. Una vez que estén todas encendidas permanecerán así 5 s. Transcurrido dicho tiempo se apagarán todas y volverá a iniciarse el ciclo.

6. Se dispone de un motor, conectado a las salidas S1 y S2, y queremos controlarlo con la teclas d (giro a la derecha), i (giro a la izquierda) y p (parada). Realizar el programa para controlar dicho motor.

7. Programar el funcionamiento de un semáforo para un paso de peatones. Las conexiones a la placa controladora serán las siguientes:

Peatones		Coches			Tiempo
Rojo	Verde	Rojo	Ambar	Verde	
S5	S4	S3	S2	S1	

8. Realizar un programa para controlar, con la teclas correspondientes del teclado numérico, el movimiento (derecha, izquierda, adelante y atrás) del vehículo que utilizamos en el buscador de luz. Dispone de dos motores conectados a las salidas S1 y S2 el motor derecho y S3, S4 el motor izquierdo.

9. Indicar que hace el siguiente programa:

```
para actividad2
leerentradas
si :el = 1 [salidas 1 espera 480 salidas 0]
actividad2
fin
```


10. Al accionar la tecla a se encenderá una luz, conectada a la salida S1. Si la accionamos por 2ª vez se encenderá otra luz conectada a la salida S2. Por último, al accionarla por 3ª vez se apagarán las dos luces y se vuelve a iniciar el ciclo.

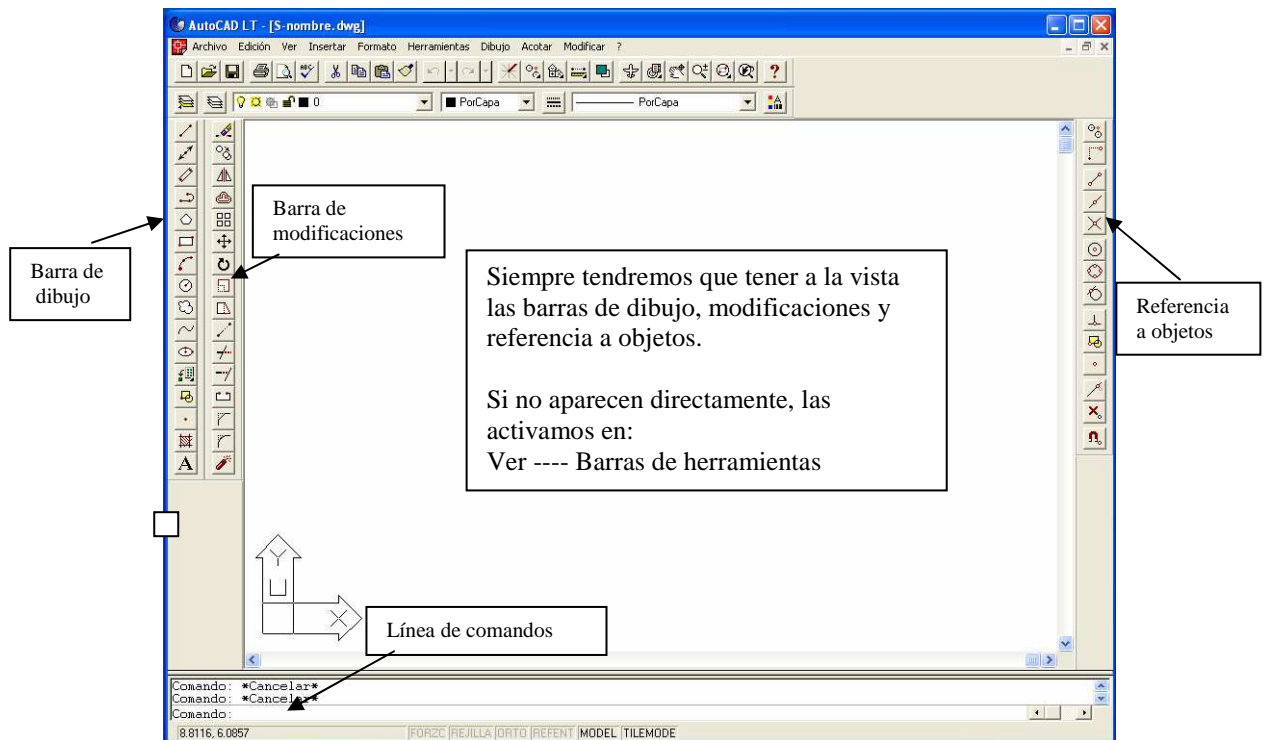
11. Al activar las entradas 1 ó 2 se debe poner en marcha un motor (en S1). Posteriormente, al activar la entrada 3 se para el motor y se enciende una luz (en S2). Por último, al activar la entrada 4 se apagará la luz.

12. Realizar un programa para controlar la apertura y cierre de la puerta de un aparcamiento, teniendo en cuenta las siguientes condiciones:

- a) El motor que se encarga de abrir y cerrar la puerta se conecta a las salidas S1 y S2.
- b) La puerta se abre al accionar un pulsador exterior conectado a E1 o un pulsador interior conectado a E2.
- c) Disponemos de un interruptor final de carrera (en E3), para detectar cuando la puerta está completamente abierta.
- d) Disponemos de un final de carrera (en E4) que detecta cuando está completamente cerrada.

13. Realizar un programa para controlar el funcionamiento de un ascensor de tres pisos, teniendo en cuenta las siguientes condiciones:
- El motor que se encarga de subir y bajar la cabina está conectado a las salidas 1 y 2.
 - Disponemos de tres pulsadores, en cada uno de los pisos para llamar al ascensor (conectados a E1 el más bajo, a E2 el intermedio y a E3 el superior).
 - Existen tres sensores, uno en cada piso, que suministran un 1 cuando la cabina pasa frente a ellos (conectados a E4 el del piso más bajo, a E5 el del piso intermedio y a E6 el del piso superior).
- (Utilizar simascensor para comprobar su funcionamiento)
14. Realizar un programa para controlar el funcionamiento de un vehículo rastreador (sigue una línea negra dibujada sobre un fondo blanco). El vehículo cuenta con dos motores conectados a las salidas S1, S2, S3 y S4 y dos sensores, situados inicialmente sobre la línea negra, que suministran un 1 cuando el color es negro y un cero cuando es blanco. (Utilizar simrastreador).

AutoCAD LT



INTRODUCCIÓN DE COORDENADAS.- En autocad, se dispone de los siguientes métodos para indicar puntos sobre el dibujo:

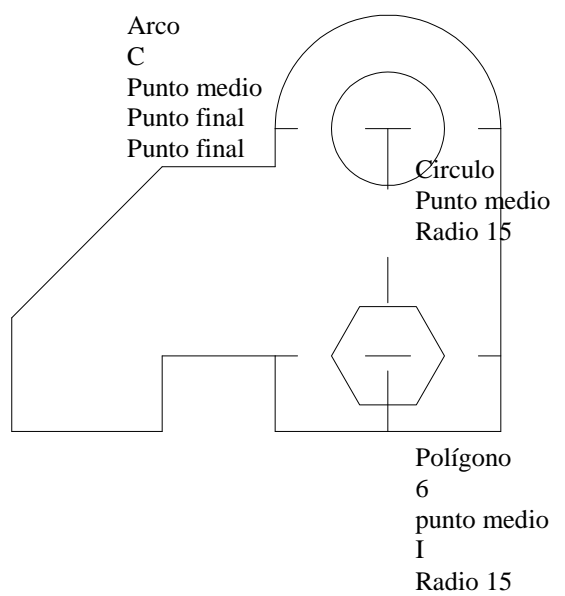
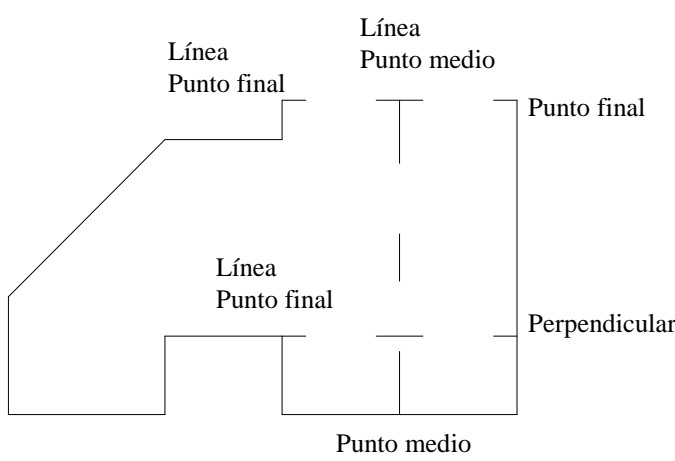
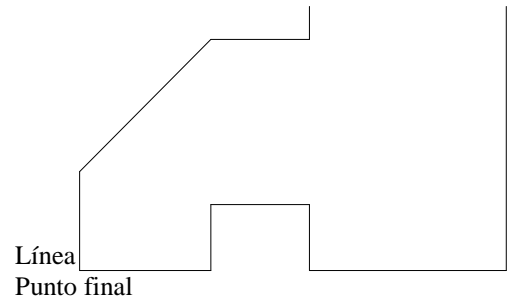
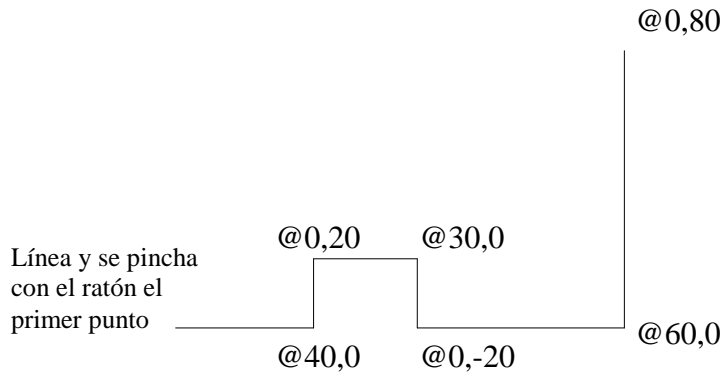
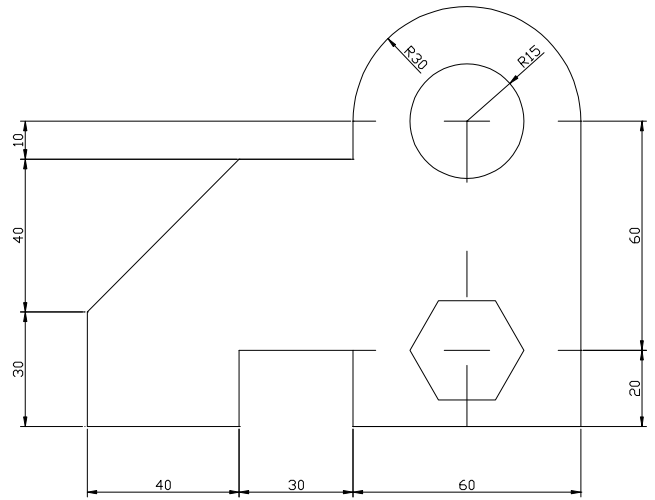
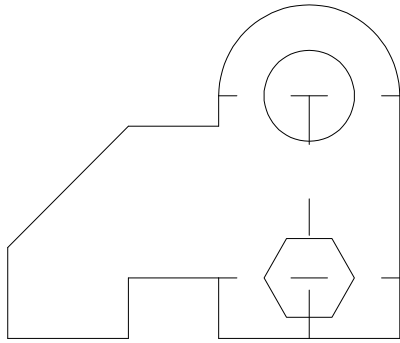
Coordenadas cartesianas absolutas.- Los valores que se indican están referidos al origen de coordenadas, punto (0,0). Para introducir un punto se indica el valor de X y, con una coma entre ellos, el valor de Y, es decir: X,Y.

Coordenadas cartesianas relativas.- Permiten especificar puntos relativos al último punto introducido. En este caso las coordenadas deben ir precedidas por el carácter @, es decir: @X,Y.

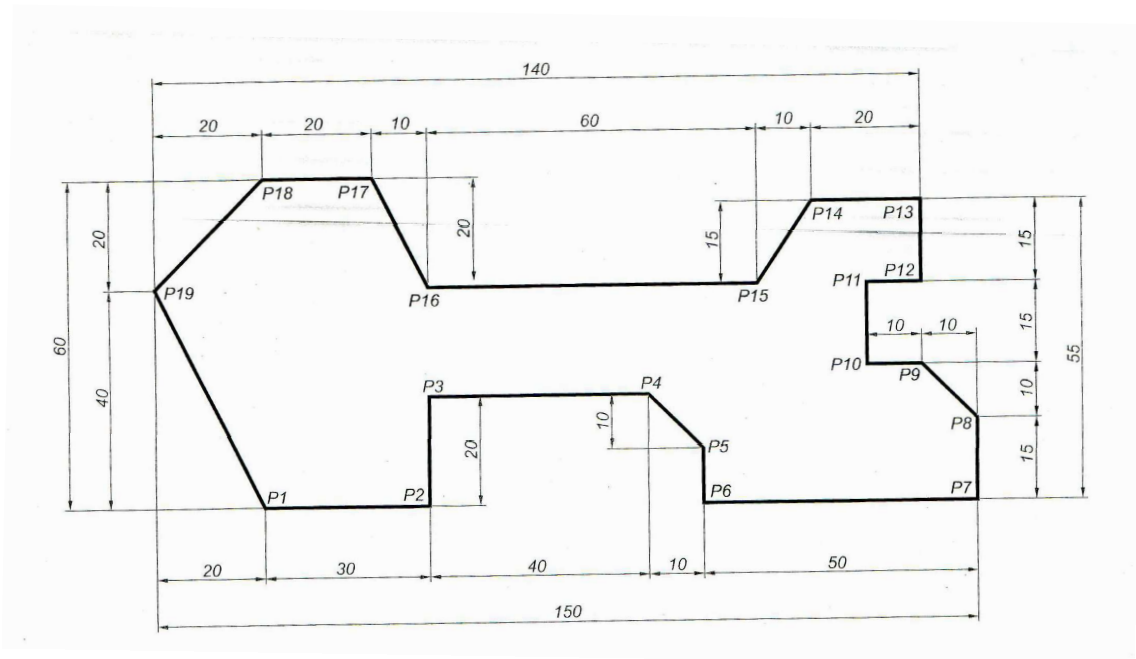
Coordenadas polares relativas.- El formato es @L< ϕ donde L indica distancia y ϕ , ángulo.

REFERENCIA A OBJETOS.- Otra manera de definir puntos es utilizando la referencia a objetos: punto final, punto medio, intersección, centro, cuadrante, perpendicular, cercano, etc.

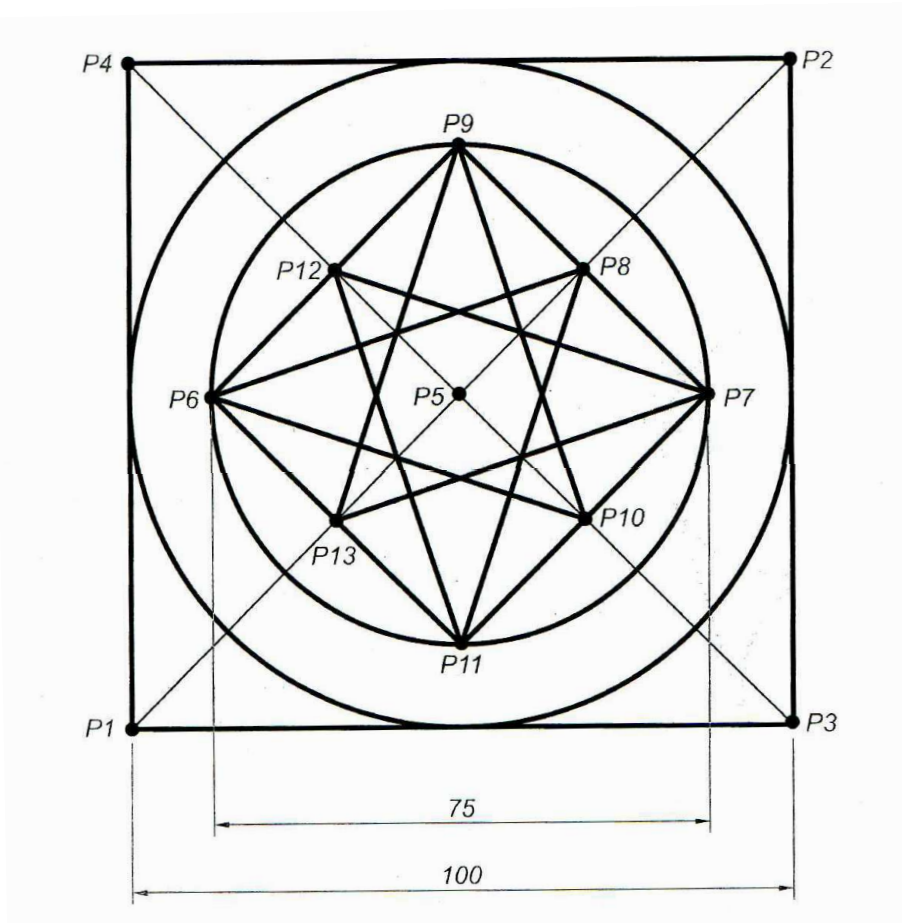
DIBUJO 1.



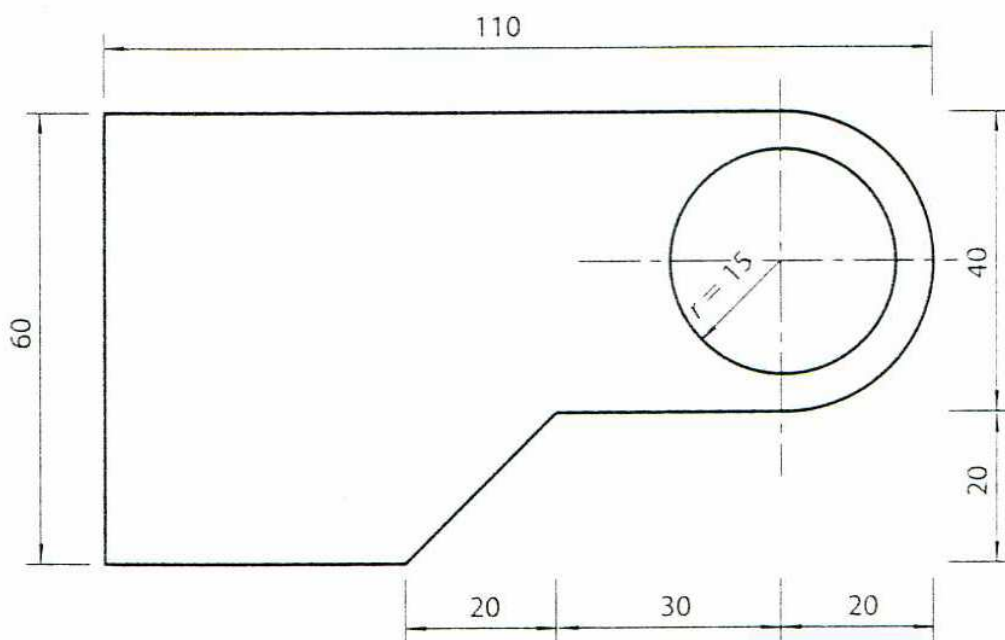
DIBUJO 2



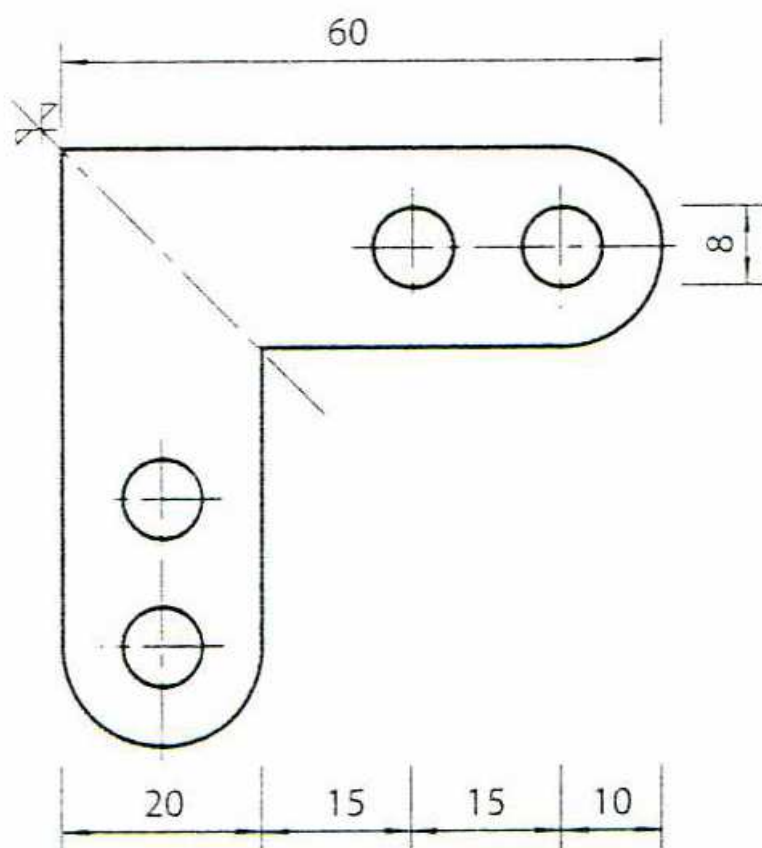
DIBUJO 3



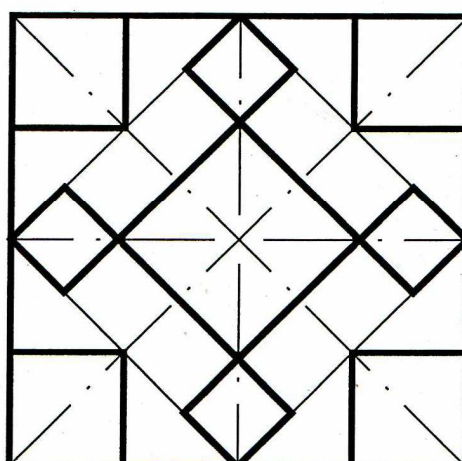
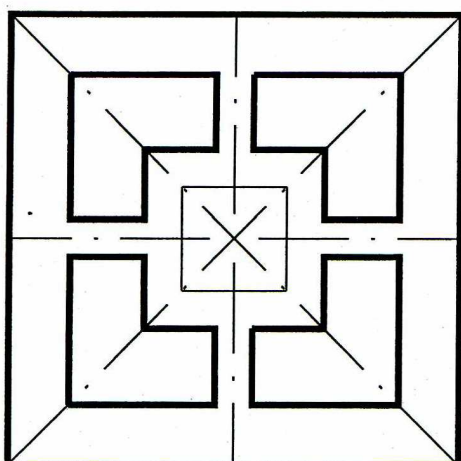
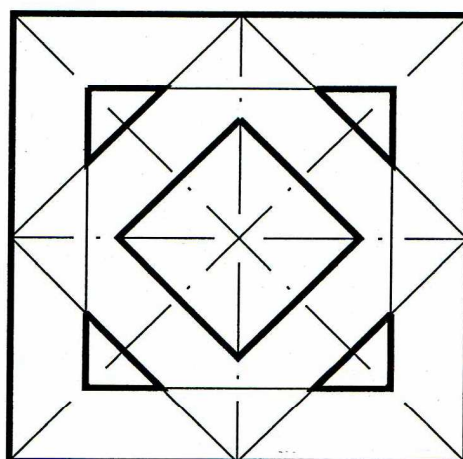
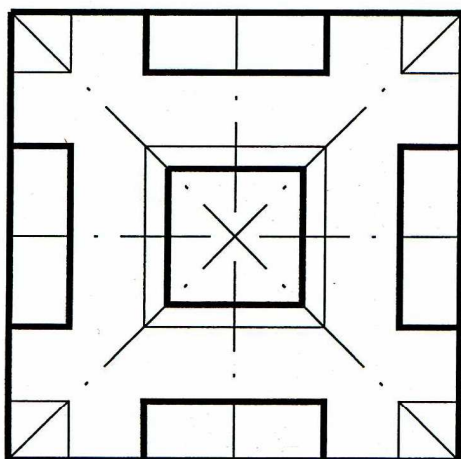
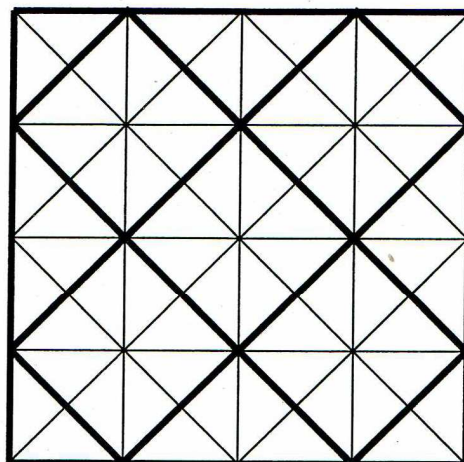
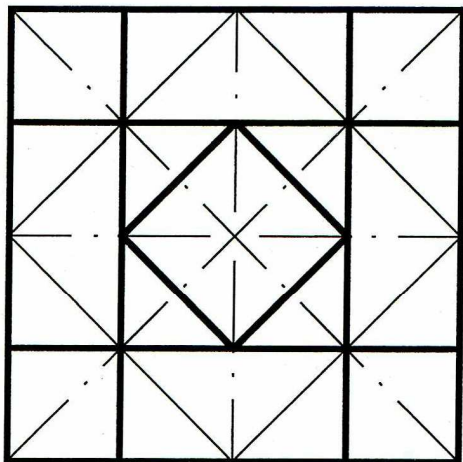
DIBUJO 4



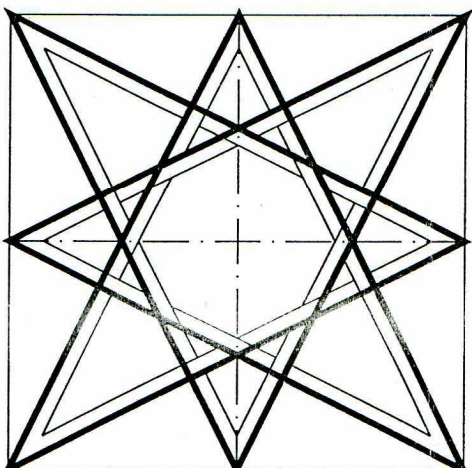
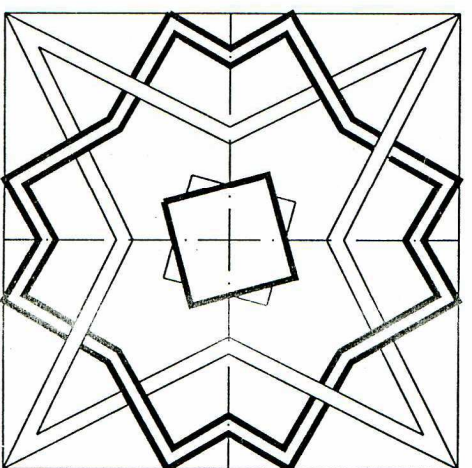
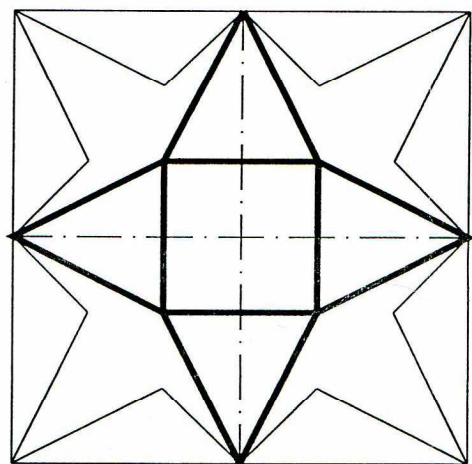
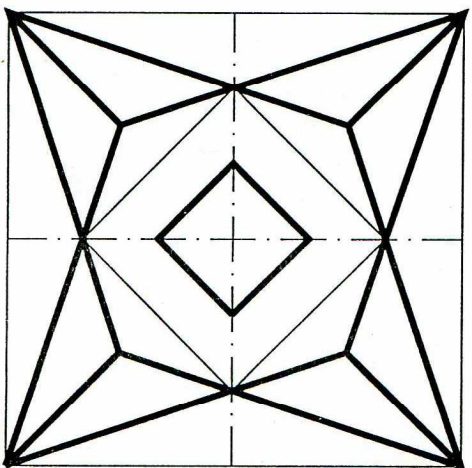
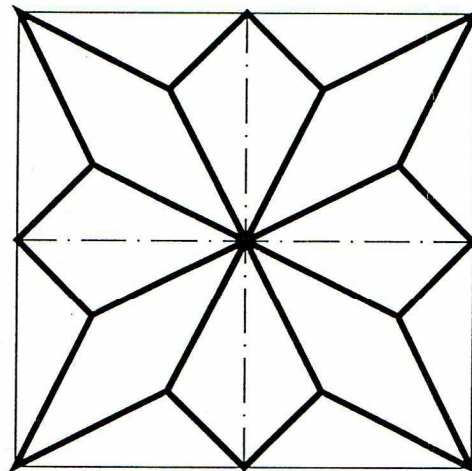
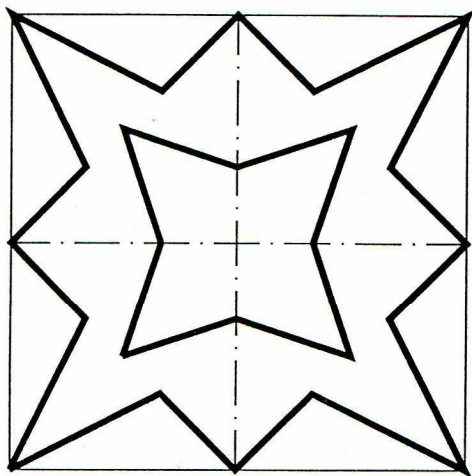
DIBUJO 5



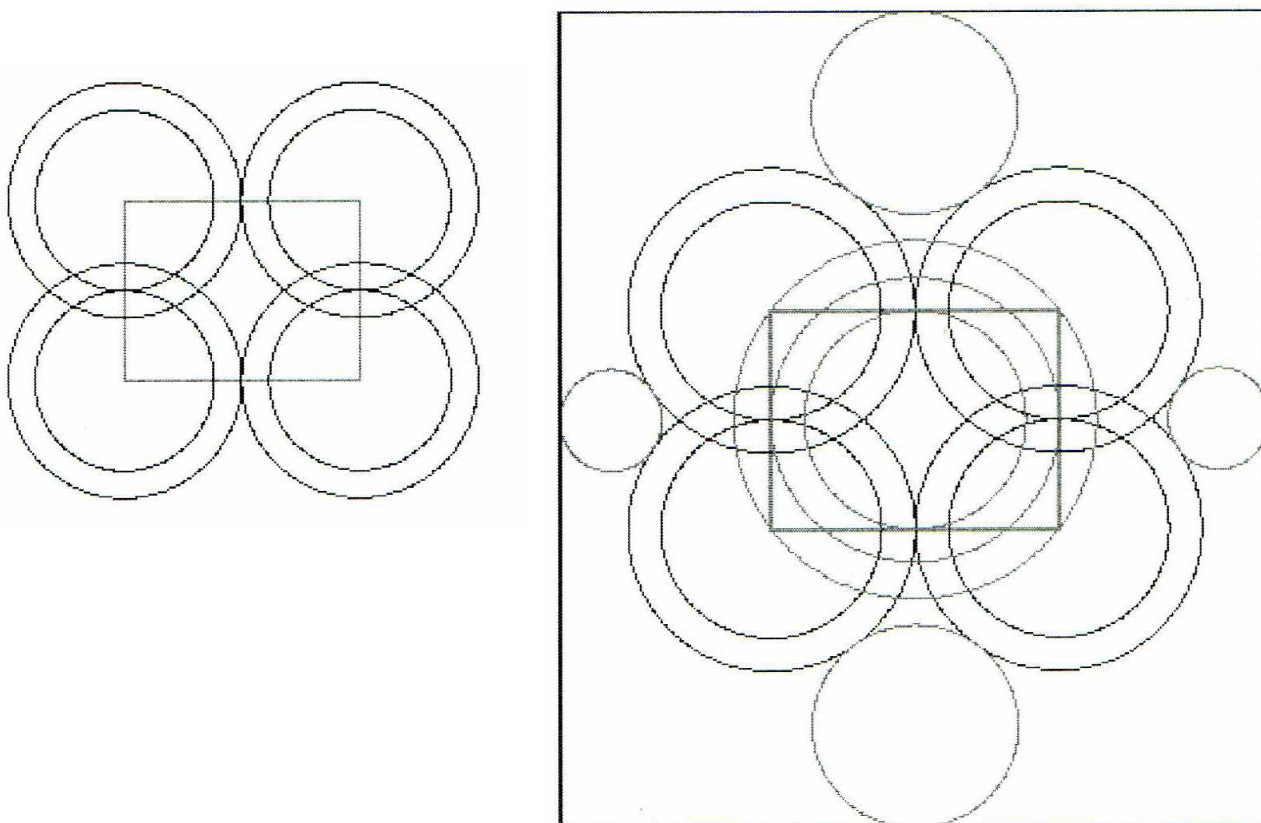
DIBUJO 6



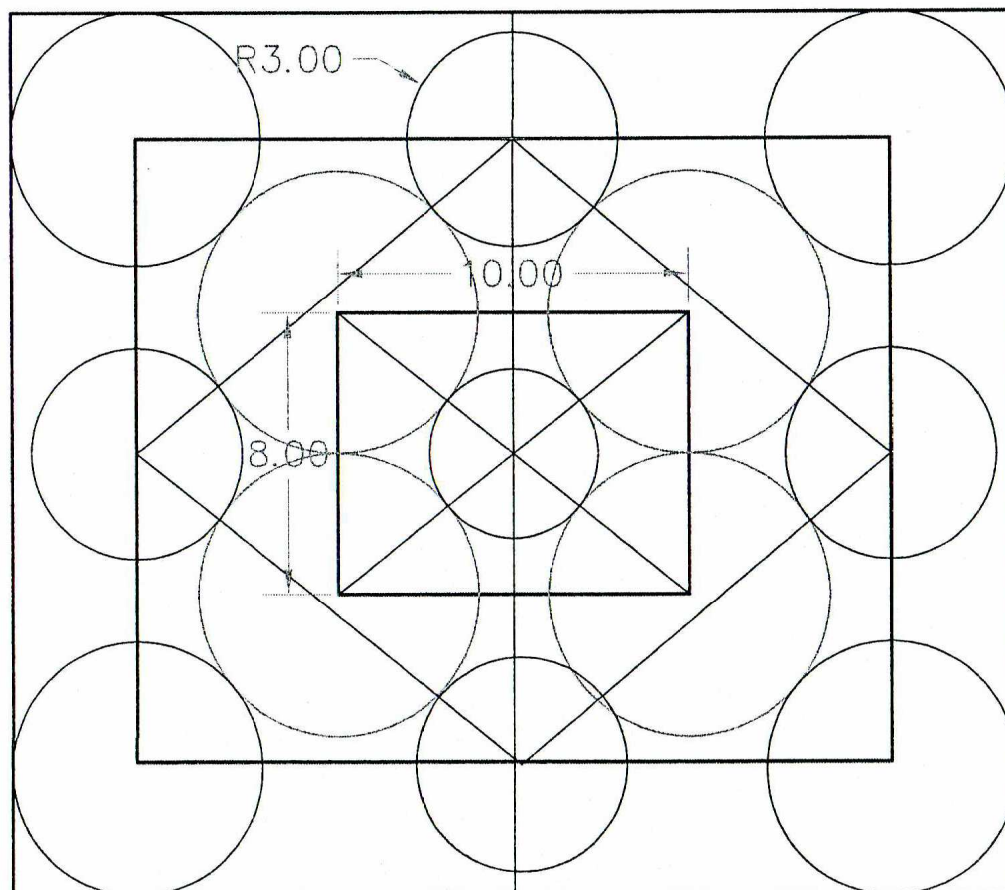
DIBUJO 7



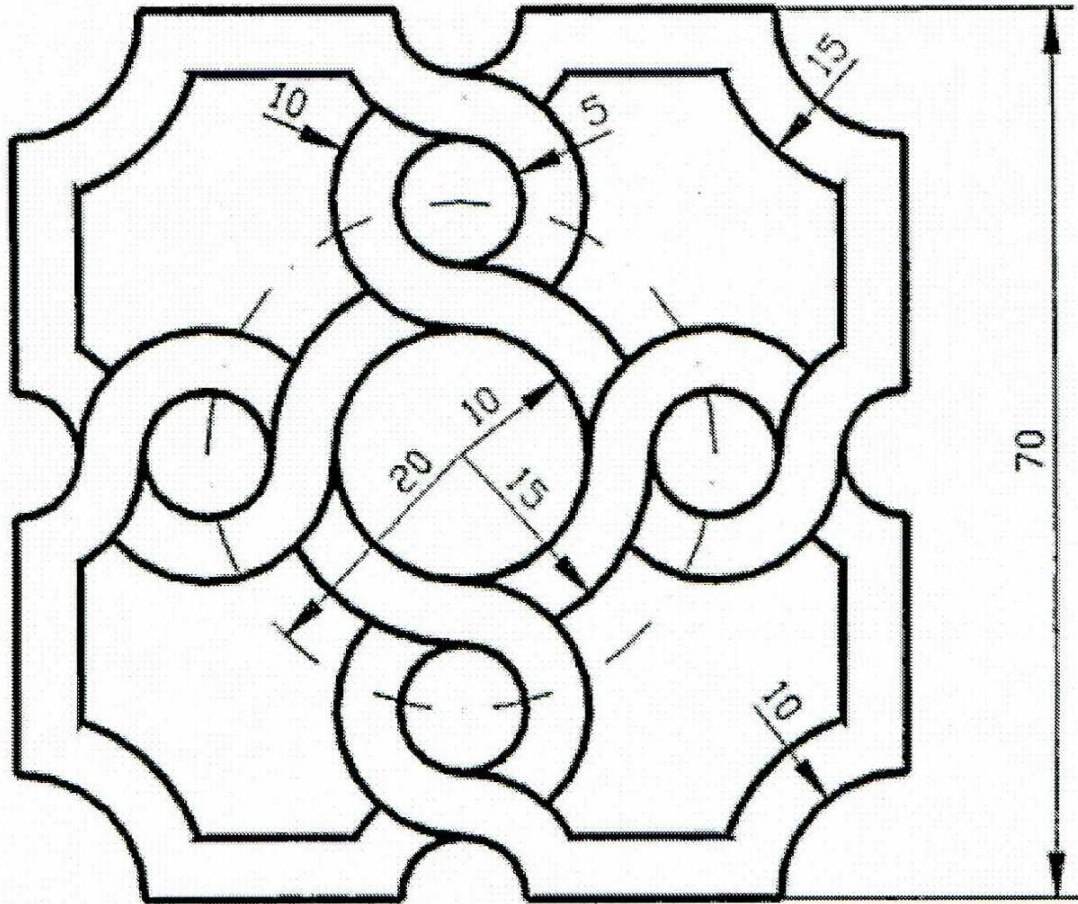
DIBUJO 8



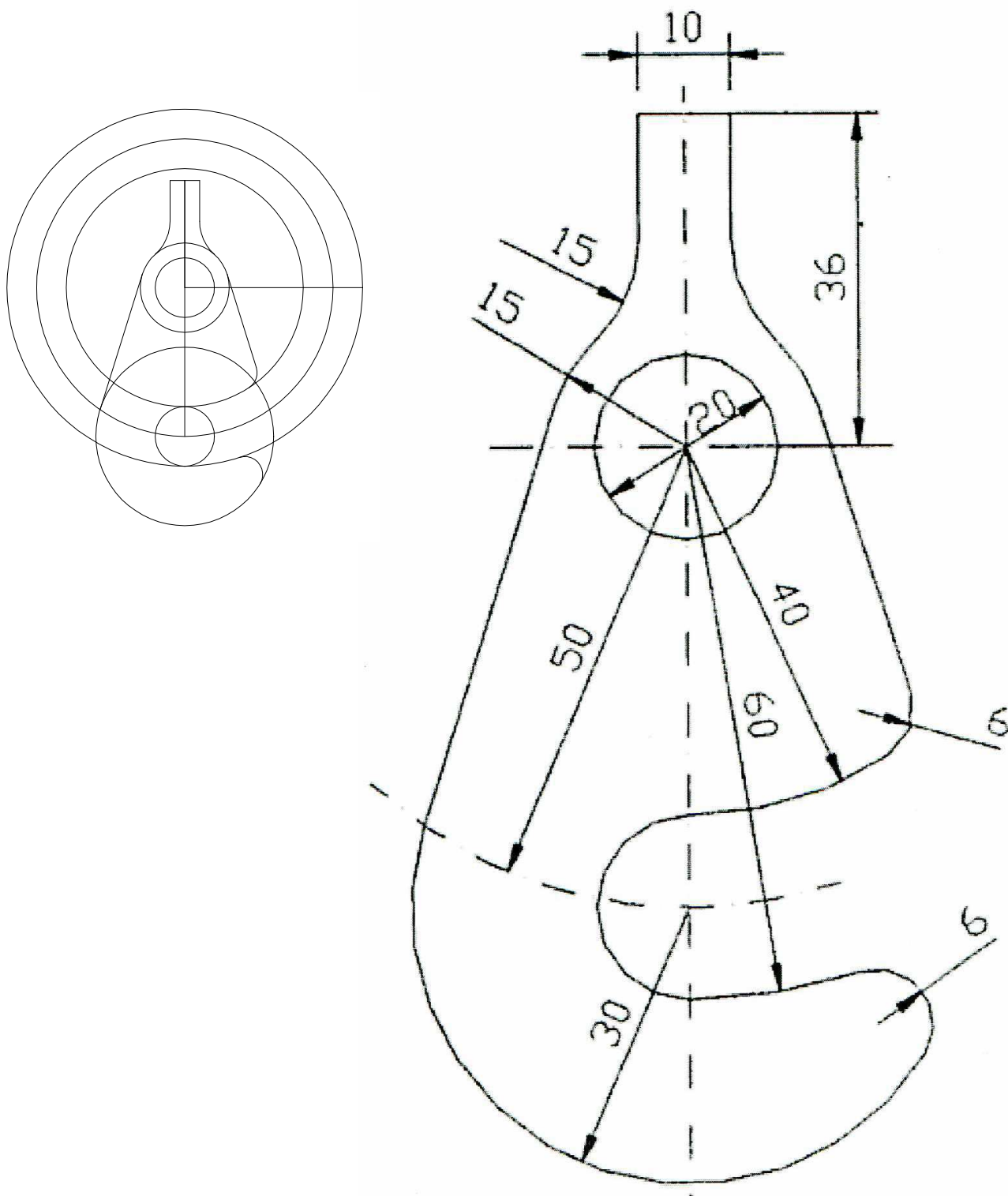
DIBUJO 9



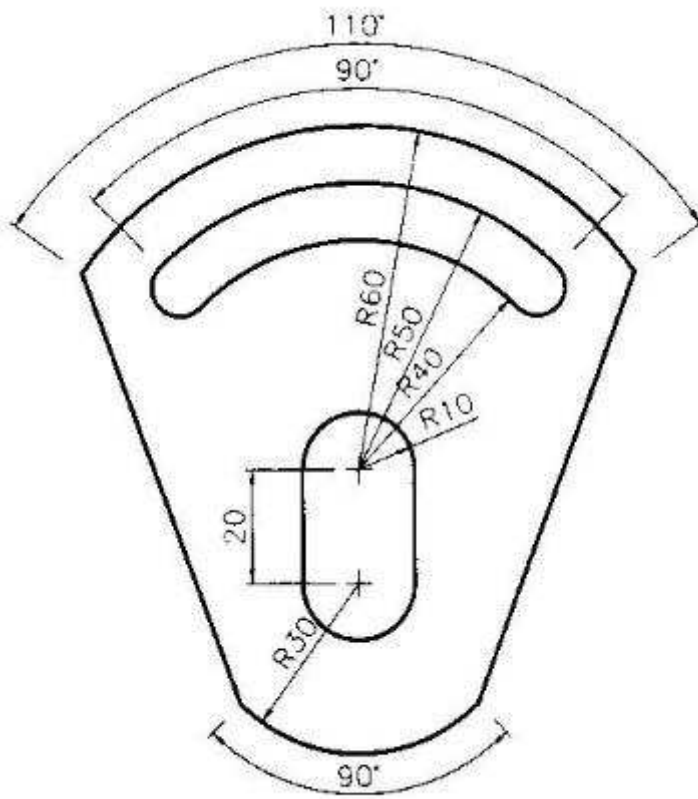
DIBUJO 10



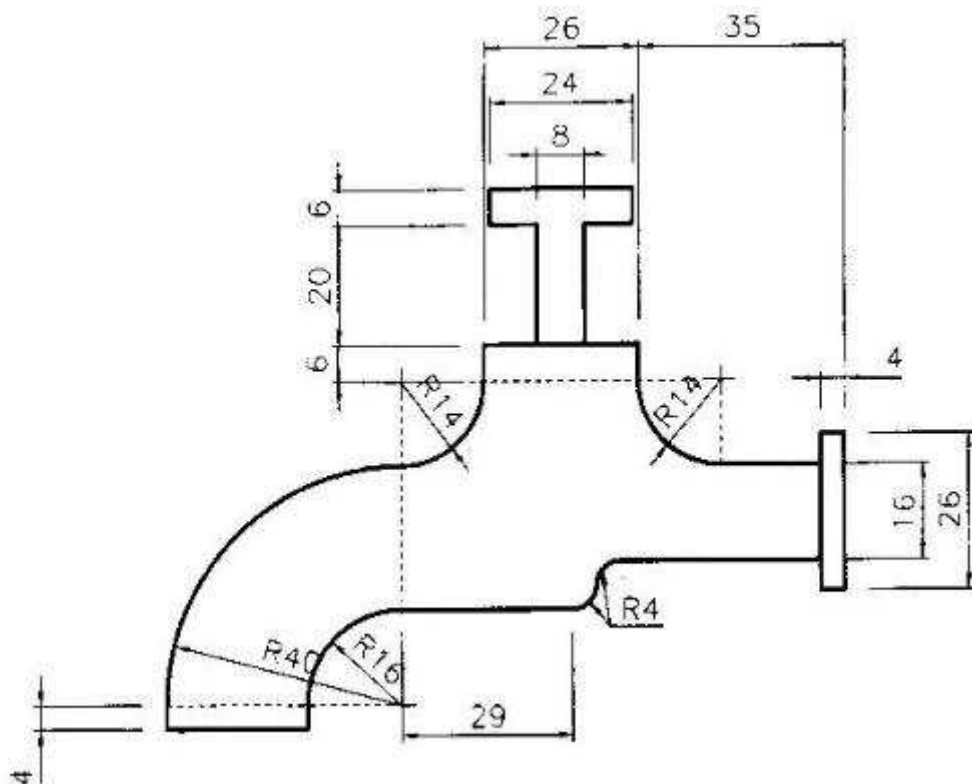
DIBUJO 11



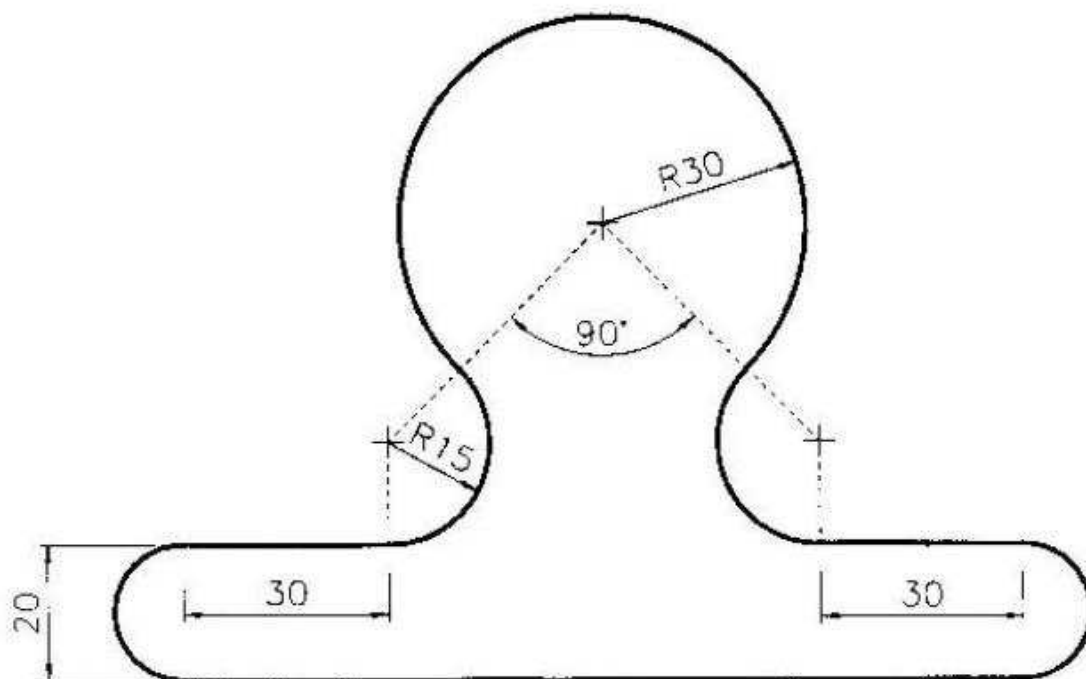
DIBUJO 12



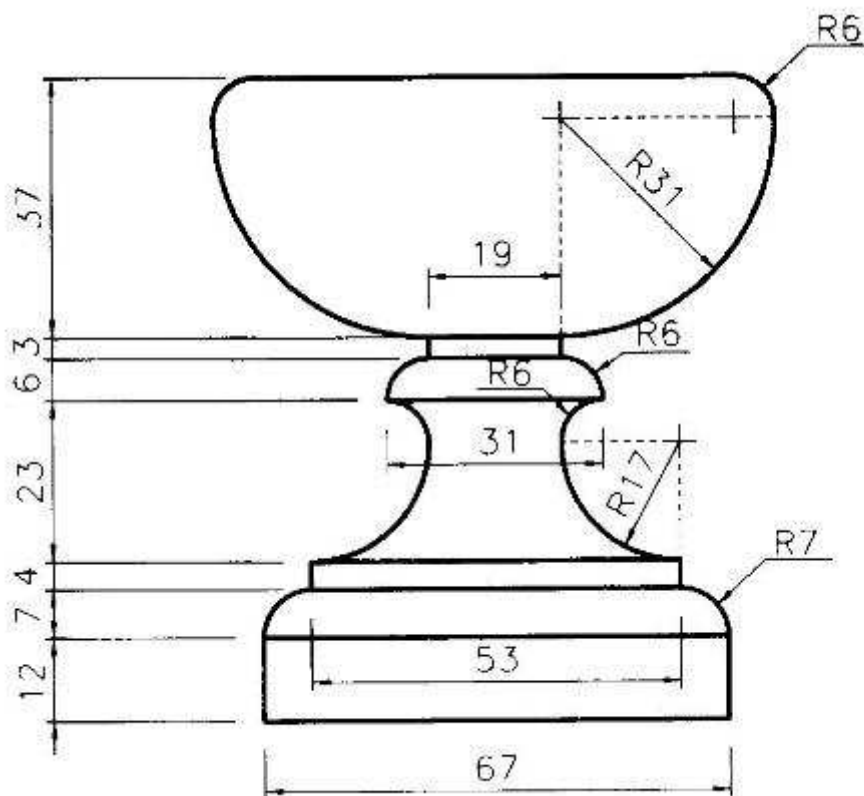
DIBUJO 13



DIBUJO 14



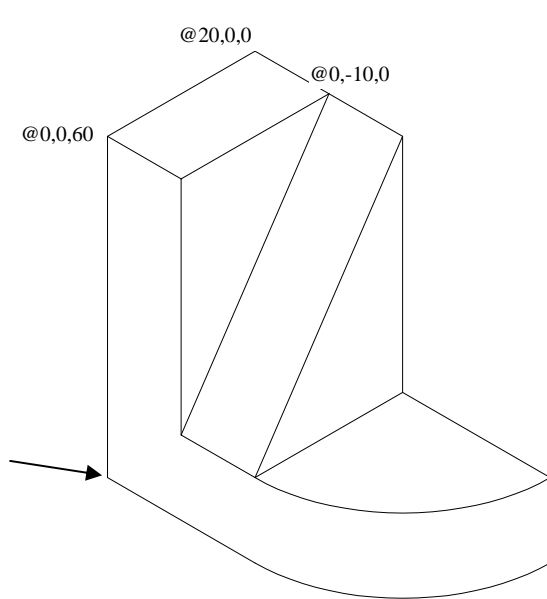
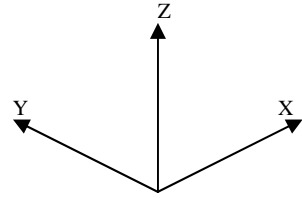
DIBUJO 15



DIBUJO 16

Seleccionar Ver---Punto de vista 3D-----Isométrico SO

Para realizar el dibujo tenemos que ir introduciendo tres coordenadas



Este arco lo hacemos copiando el de arriba

