UNIDAD DIDÁCTICA XVII

Son autores de esta obra colectiva:

Ángel Sánchez Solanilla
Andrés Sánchez Otín
Máximo Bolea Campos

Coordinación pedagógica:

Carmen Candioti

MINISTERIO DE EDUCACIÓN Y CIENCIA
SECRETARÍA DE ESTADO DE EDUCACIÓN
Programa de Nuevas Tecnologías de la Información y de la Comunicación

I.- INTRODUCCIÓN

I.- INTRODUCCIÓN

PRESENTACIÓN

En esta unidad, a través de la construcción y programación de un molinillo, se estudiará el uso y control de entradas digitales y salidas analógicas presentando, además, en la sección "Manos a la Obra", una actividad de control de una vagoneta mediante sensores digitales. En la sección "Entre Máquinas y Herramientas" se abordará el control de una grúa, con el propósito de iniciar el estudio del uso y control de sensores analógicos. Desde el apartado "Con Nuestros Alumnos y Alumnas", se mostrará la metodología más apropiada para transmitir a nuestro alumnado ideas y destrezas para confeccionar un programa y comprender su estructura diseñando lo que llamaremos "el pensamiento de la máquina".

1.- OBJETIVOS

En esta unidad didáctica, se pretende que el profesorado consiga los siguientes objetivos:

- Comprender las funciones de las salidas analógicas y desarrollar capacidades para programar el control de las mismas.

- Usar y controlar entradas analógicas. (El potenciómetro).
- Crear procedimientos comprensibles por el alumnado para que pueda avanzar en el manejo de Logo como entorno de reflexión.
- Utilizar técnicas adecuadas al trabajo en grupo de nuestros alumnos y alumnas.

2.- CONTENIDOS

I.- INTRODUCCIÓN

II.- FUNDAMENTOS DE CONTROL

1.- Semáforo con pulsador.
1.1.- Conexiones.
1.2.- Fases del ciclo.

III.- MANOS A LA OBRA.
1.- Salidas analógicas.

2.- El molinillo.
2.1.- Planteamiento del problema.

2.2.- Las conexiones.
2.2.a.- Primeras pruebas.
2.2.b.- Usando variables.
2.3.- Los procedimientos.
2.3.a.- Mejorando los procedimientos.

IV.- CON NUESTROS ALUMNOS Y ALUMNAS

Estrategias para presentar en el aula contenidos de programación en Logo

1.- Programas estructurados.
1.1.- Programación

2.- Las variables.
2.1.- Variables locales.
2.2.- Variables globales.
2.3.- Metodología para la presentación de las variables locales.
2.4.- Metodología para la presentación de las variables globales.

3.- Sentencias condicionales
3.1.- Metodología.
3.2.- Los filtros.
3.3.- Los menús.

V.- ENTRE MÁQUINAS Y HERRAMIENTAS

1.- Control de una grúa.
1.1.- Conexiones.
1.2.- Control de la grúa.
1.3.- Propuesta de trabajo. Mejora del sistema.

3.- CONOCIMIENTOS PREVIOS

El profesorado debe haber afianzado sus capacidades para:

- Controlar entradas digitales.
- Controlar salidas digitales.
- Conectar circuitos en paralelo.
- Programar en LOGO.
- Aplicar al control sus conocimientos de resistencia eléctrica.

II.- FUNDAMENTOS DE CONTROL

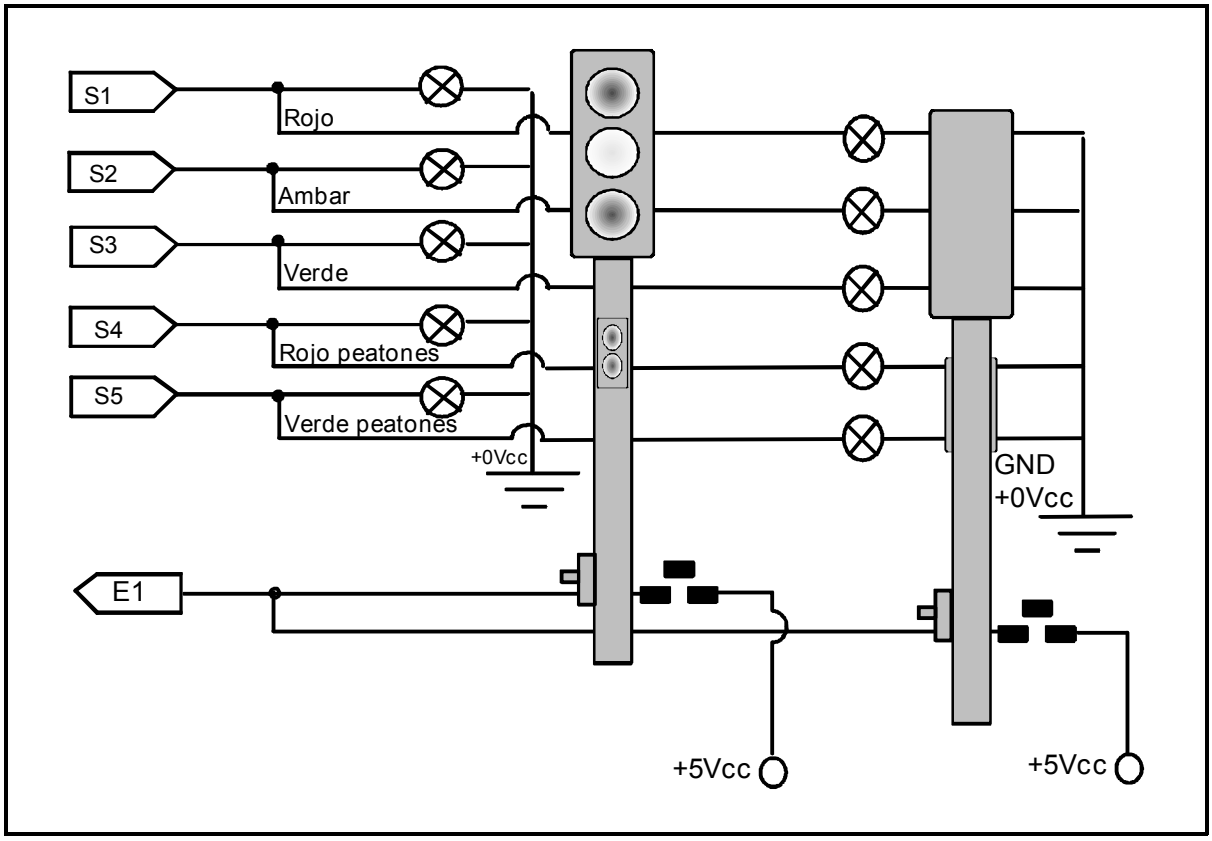
II.- FUNDAMENTOS DE CONTROL

1.- SEMÁFORO CON PULSADOR

En la unidad didáctica anterior, se presentó una actividad al final del apartado "Entre máquinas y herramientas"; dicha actividad consiste en realizar el control de un sistema de semáforos para vehículos y peatones. aunque quienes sigan este curso hayan diseñado soluciones, se presenta a continuación otra solución posible, donde se fundamentan los procedimientos y decisiones adoptados.

Recuérdese que había que colocar semáforos que permitan el paso de peatones, cada cierto tiempo, en el punto medio de la longitud de una calle. Para el control, deberá tenerse en cuenta además que los peatones puedan accionar en cada acera el pulsador que les dará prioridad (al accionarlo iniciará el ámbar para los automóviles).

1.1.- CONEXIONES



Las luces rojas de los semáforos de los vehículos se encienden y apagan al mismo tiempo. Por eso, se han conectado en paralelo a la salida S1 y a tierra GND.

Otro tanto sucede con las luces amarillas de los semáforos de los vehículos, que se encienden y apagan al mismo tiempo. Así, se han conectado en paralelo a la salida S2 y a tierra GND.

Análogamente, las luces verdes de los semáforos de los vehículos se encienden y apagan al mismo tiempo. Por eso se han conectado en paralelos a la salida S3 y a tierra GND.

La luz roja del semáforo para peatones se ha conectado a S4 y GND. La activación de esta luz deberá coincidir con la de la luz verde o ámbar para los automóviles. (Podría ahorrarse esta salida digital llevando la línea simultáneamente a S2 y S3, ámbar y verde de los automóviles).

La luz verde del semáforo para peatones se ha conectado a S5 y GND. La activación de esta luz deberá coincidir con la de la luz roja para los automóviles. (Podrían haberse conectado en una misma salida digital).

Basta con que un peatón pulse su botón para que el sistema responda; por ello, se han colocado los dos sensores digitales en paralelo, conectados a la entrada digital E1 y a +5Vcc de la controladora.

1.2.- FASES DEL CICLO

FASE 1 Amarillo para automóviles y rojo para peatones. Supongamos que dura 5 segundos, tiempo necesario para que los vehículos reaccionen frenando. Durante dicho período no se tendrá en cuanta la pulsación de los botones, ya que la luz verde para peatones viene a continuación.

FASE 2 Rojo para automóviles y verde para peatones. Supongamos que dura 10 segundos, tiempo necesario para que crucen los peatones. En ese tiempo, no se considerará la pulsación de petición de paso, ya que los peatones tienen la prioridad.

FASE 3 Verde para los vehículos y rojo para peatones. Supongamos que dura 40 segundos, salvo la petición de los viandantes por pulsación del botón haga que su duración sea menor. Tanto si transcurren los 40 segundos como si se pulsa un botón, habrá de reiniciarse el ciclo con la necesaria luz amarilla, para el frenado de los vehículos, que precede a la concesión de paso a los peatones.

1) La programación de la fase 1 no entraña dificultad. Para activarla las salidas digitales tendrán que ponerse a:

S8 = 0 No influye
S7 = 0 No influye
S6 = 0 No influye
S5 = 0 Apagado verde peatones
S4 = 1 Encendido rojo peatones
S3 = 0 Apagado verde coches
S2 = 1 Encendido ámbar coches
S1 = 0 Apagado rojo coches

La orden que la activa esta fase será:

```
SALIDA VALORBIN "S8_S7_S6_S5_S4_S3_S2_S1_
SALIDA VALORBIN "00001010
```

El procedimiento apropiado podría escribirse como sigue:

```
PARA FASE1
SALIDA VALORBIN "00001010   ESPERA  500 ;   Activa la fase y espera 5s.
FIN
```

2) La programación de la fase 2 tampoco supone ninguna dificultad. Para activarla, las salidas digitales tendrán que ponerse a:

S8 = 0 No influye
S7 = 0 No influye
S6 = 0 No influye
S5 = 1 Encendido verde peatones
S4 = 0 Apagado rojo peatones
S3 = 0 Apagado verde coches
S2 = 0 Apagado ámbar coches
S1 = 1 Encendido rojo coches

La orden que la activa esta fase será:

```
SALIDA VALORBIN "S8_S7_S6_S5_S4_S3_S2_S1_
SALIDA VALORBIN "00010001
```

El procedimiento apropiado puede escribirse como sigue:

```
PARA FASE2
SALIDA VALORBIN "00010001  ESPERA  1000 ;   Activa la fase y espera 10s.
FIN
```

3) La programación de la fase 3 ha de contar con los pulsadores conectados a la entrada E1. Para activar las luces correspondientes a las salidas digitales oportunas, tendrán que ponerse a:

S8 = 0 No influye
S7 = 0 No influye
S6 = 0 No influye
S5 = 0 Apagado verde peatones
S4 = 1 Encendido rojo peatones
S3 = 1 Encendido verde coches
S2 = 0 Apagado ámbar coches
S1 = 0 Apagado rojo coches

La orden que activa esta fase será:

```
SALIDA VALORBIN "S8_S7_S6_S5_S4_S3_S2_S1_
SALIDA VALORBIN "00001100
```

A continuación y durante el resto de la fase (máximo 40 sg) se tendría que leer el estado del sensor digital conectado a E1, con objeto de interrumpirla. Esta operación la desarrollaremos en el procedimiento PULSADO. A la espera de desarrollar este procedimiento, el programa apropiado para la fase 3 puede escribirse como sigue:

```
PARA FASE3
SALIDA VALORBIN "00001100 PULSADO ;Activa la fase y pasa a PULSADO.
FIN
```

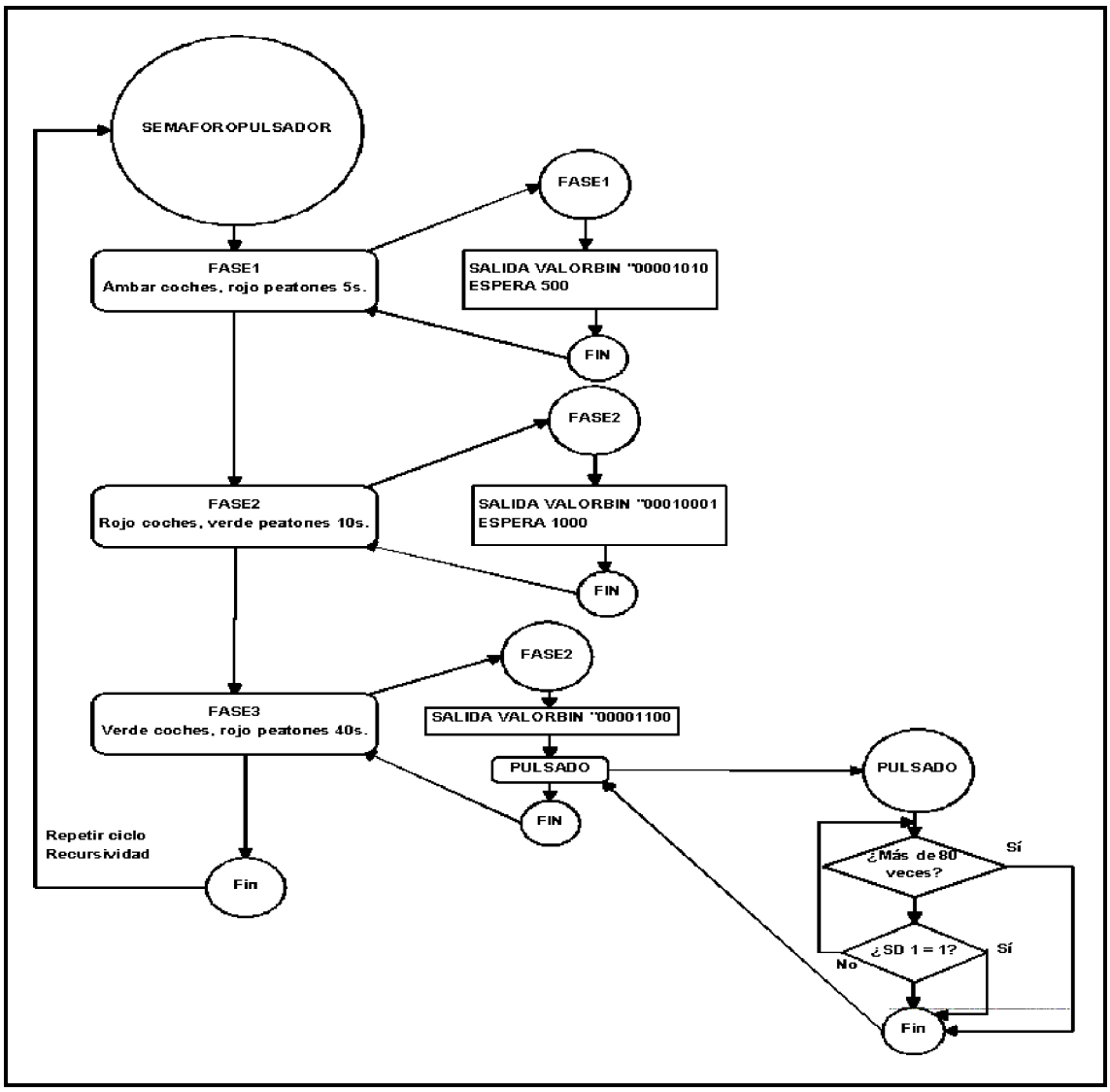
Para desarrollar el procedimiento PULSADO, tendremos en cuenta que la controladora invierte un tiempo en chequear el sensor digital. En realidad, invierte aproximadamente un segundo para chequearlo dos veces consecutivas. Podríamos decidir que realizar el chequeo 80 veces y dispondríamos de la espera necesaria (40 segundos); no obstante, la frecuencia deberá ajustarse experimentalmente. Si en algún chequeo el SD_1 devolviera "1" (algún pulsador accionado), se ordenaría la interrupción del procedimiento PULSADO, con lo que se regresaría al procedimiento que lo llamó: FASE3, que al no tener más instrucciones concluiría.

El procedimiento PULSADO puede escribirse así:

```
PARA PULSADO
REPITE 80 [SI SD 1 = 1 [ALTO] ] ;      Repite 80 veces el chequeo, pero si  el SD_1
                                       vale "1" se interrumpe (ALTO).
FIN
```

El programa que realiza cíclicamente (recursivamente) las tres fases es el siguiente:

```
PARA SEMAFOROPULSADOR

FASE1;      Se realiza la fase 1ª (Ámbar para los coches y rojo para peatones,  -5
            segundos- )

FASE2;      Se realiza la fase 2ª (Rojo para los coches y verde para peatones,  -10
            segundos- )

FASE3;      Se realiza la fase 3ª (Verde para los coches y rojo para peatones, la fase
            termina si se activa un pulsador o si transcurren 40 segundos )

SEMAFOROPULSADOR;          Se reinicia el ciclo, llamada recursiva

FIN
```



El programa completo quedaría:

```
PARA FASE1
SALIDA VALORBIN "00001010   ESPERA  500
FIN
PARA FASE2
SALIDA VALORBIN "00010001  ESPERA  1000
FIN
PARA PULSADO
REPITE 80 [SI SD 1 = 1 [ALTO] ]
FIN
PARA FASE3
```

```
SALIDA VALORBIN "00001100 PULSADO
FIN
PARA SEMAFOROPULSADOR
FASE1
FASE2
FASE3
SEMAFOROPULSADOR
FIN
```

ACTIVIDAD RECOMENDADA

Realiza la construcción del semáforo anterior y haz las conexiones pertinentes para mejorar el diseño, ya que se pueden ahorrar salidas digitales. Reescribe los programas para esta situación más económica. Conecta el sistema de semáforos a la controladora. Enciende el ordenador, carga Windows, Win-Logo y el micromundo de control. Edita en la ventana de trabajo el programa realizado, ejecútalo, afínalo mediante pruebas y archívalo.

Elabora una breve memoria del trabajo realizado y envíalo a la tutoría.

III.- MANOS A LA OBRA

III.- MANOS A LA OBRA

En esta nueva unidad del curso "Del Clavo al Ordenador" se estudiará el uso y el control de salidas analógicas. Recordemos que una salida analógica permitirá controlar la velocidad de giro de un motor o la intensidad luminosa de una bombilla.

Veáse vídeo de apoyo de la unidad.

1.- SALIDAS ANALÓGICAS

Las salidas analógicas de la controladora, como sabemos, son las marcadas como S9-S10, correspondientes al actuador situado en la posición número 5.

Las primitivas que controlan las salidas analógicas son:

M5 "D gira en un determinado sentido
M5 "I gira en sentido contrario al anterior

M5 "P se detiene el motor

ACELERA :N

Es la orden que realmente pone de manifiesto la potencia de esta salida analógica. :N es un número comprendido entre 0 y 15. Lo que hace la controladora es suministrar un voltaje comprendido entre 0 y 6, 5 voltios en función del valor de :N.

Ejemplos del uso de ACELERA: ACELERA 0 (mínima velocidad)

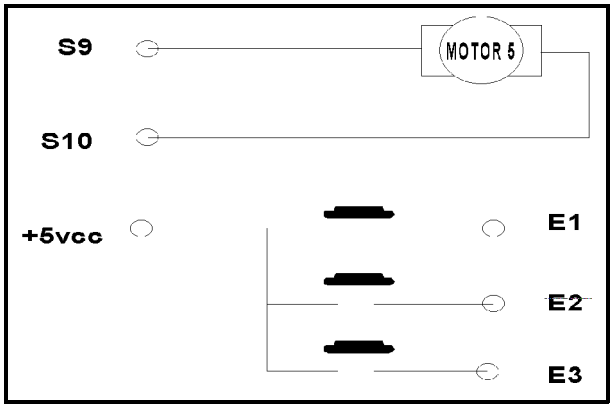
ACELERA 15 (máxima velocidad)
ACELERA 3

2.- EL MOLINILLO

2.1.- PLANTEAMIENTO DEL PROBLEMA

Como ejercicio para el control de las salidas digitales, planteamos controlar un molinillo mediante tres sensores digitales. Uno de ellos sirve para la puesta en marcha, otro para acelerar la velocidad de giro y el tercero para ir frenando el giro del molinillo.

2.2.- LAS CONEXIONES.



El motor que hace girar el molinillo se conecta a S9-S10, salidas analógicas, que permiten el control de la velocidad del motor, mediante la variación del voltaje suministrado a través de la primitiva ACELERA. Los tres sensores digitales se conectan de la forma ya conocida y su control es el de siempre.

2.2.a.- PRIMERAS PRUEBAS

Para comprobar las conexiones podemos utilizar las órdenes de control de las salidas analógicas de un modo directo. Se carga Win-Logo con los ficheros de control y, en el área de trabajo, y se escriben y ejecutan las siguientes órdenes, para familiarizarse con ellas:

```
M5 "I
ACELERA 3
ACELERA 14
ACELERA 4
M5 "D
ACELERA 6
M5 "I
ACELERA 0
ACELERA 15
M5 "P
```

En la experimentación con estas órdenes se habrá encontrado sentido a la orden ACELERA y a M5.

Antes de usar ACELERA, las salidas analógicas S9 y S10 deben estar ya activadas. Ejemplo:

ACELERA 4 no surtiría ningún efecto de no haber ejecutado previamente M5 "D (o M5 "I). Una secuencia lógica de acciones es la siguiente: M5 "D ACELERA 3.

Pueden probarse las siguientes secuencias que aparecen en el vídeo:

```
M5 "D ACELERA 0
M5 "D ACELERA 3
M5 "D ACELERA 9
M5 "I
M5 "I ACELERA 15
M5 "P
```

2.2.B.- USO DE VARIABLES

El control de salidas analógicas es un tema adecuado para el uso de variables. La primitiva ACELERA se presta a utilizar variables.

Revisa el apartado II de la unidad XII.

Recordemos cómo declarar y asignar valor a una variable:

```
HAZ "VARIABLE VALOR
```

Variable es el nombre dado a la variable y valor su valor. Veamos algunos ejemplos, que ejecutaremos en modo directo.

```
HAZ "N 0                    A la variable N se le asigna 0 como valor
ES :N

HAZ "CC 23          A la variable llamada CC se le asigna valor 23
ES :CC

HAZ "FEDO "CASA     A la variable FEDO se le asigna como contenido la palabra
CASA
ES :FEDO

HAZ "RRE [LA META ESTA MUY LEJOS]   La variable RRE contiene la lista LA
                                    META ESTA MUY LEJOS.
ES :RRE

HAZ "Q 7                    A la variable Q se asigna valor 7
ES :Q
HAZ "Q :Q+1                 A la variable Q se le asigna el valor que ya tenía más uno.
ES :Q
```

Ejemplos de procedimientos que usan variables.

```
PARA CONTARPARA CONTAR1
HAZ "C 0ES :C
CONTAR1SI :C = 10 [ALTO]
FINHAZ "C :C+1
CONTAR1
FIN
```

Una variante de los procedimientos anteriores es la siguiente:

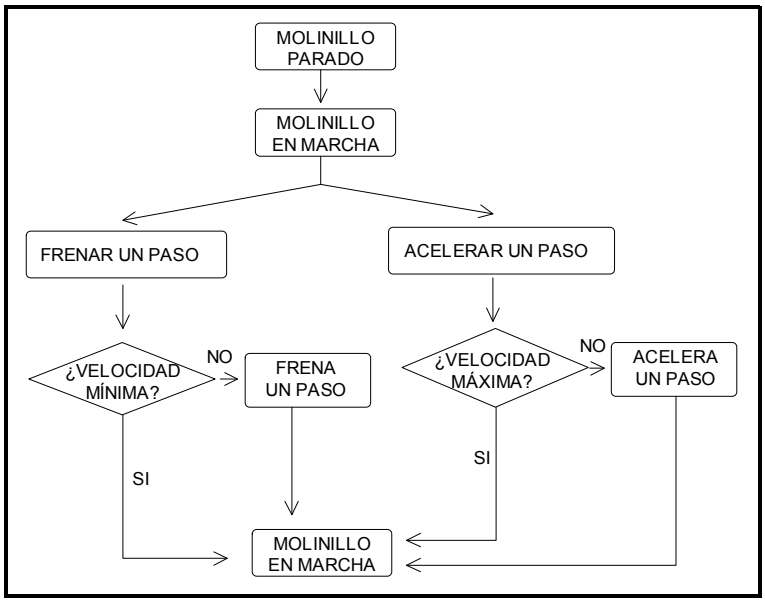
```
PARA CONTAR :C
SI :C > 10 [ALTO]
ES :C
HAZ "C :C+1
CONTAR :C
FIN
```

Escrito en el editor y pulsado el icono de intérprete, el procedimiento anterior está preparado para ser ejecutado desde el área de trabajo. Probemos escribiendo:

CONTAR 4 y pulsando [Intro].

También debemos experimentar con CONTAR 12.

2.3.- LOS PROCEDIMIENTOS



Teniendo en cuenta las condiciones del problema de controlar el molinillo y las posibilidades del lenguaje de programación, previamente debe realizarse el organigrama adecuado.

A la vista del organigrama debemos desarrollar tres procedimientos, uno que inicialice las variables y ponga en marcha el molinillo, otro para acelerar y el tercero para frenar.

Procedimiento inicial

```
PARA INICIO
HAZ "NUM 0;        inicializa la variable NUM dándole valor 0
M5 "D;             el motor lo pone en marcha
ACELERA 0;         lo acelera lo mínimo.
FIN
```

Al ejecutarse este procedimiento el molinillo realiza un movimiento muy lento, ya que ACELERA 0 implica una diferencia de potencial de 2 voltios entre las salidas analógicas.

Ahora se realizarán los procedimientos de acelerar y frenar paso a paso.

```
PARA ACELERAR_UN_PASO
HAZ "NUM :NUM + 1
ACELERA :NUM
FIN
```

En este procedimiento, puede comprobarse cómo al ejecutarse y dependiendo del valor que en ese momento tenga la variable NUM, se incrementa el valor de dicha variable en 1 y se acelera el giro del motor. Del mismo modo, pero restando 1, en el siguiente procedimiento se va frenando el giro del motor.

```
PARA FRENAR_UN_PASO
HAZ "NUM :NUM - 1
ACELERA :NUM
FIN
```

Si ejecutamos repetidamente estos procedimientos, puede suceder que el valor de variable NUM sea superior a 15; en ese momento, el ordenador emitirá un mensaje de error indicando que el valor de ACELERA debe encontrarse entre 0 y 15. Para evitarlo deberían definirse los procedimientos mediante controles internos que impidan llegar a valores superiores a 15 e inferiores a 0.

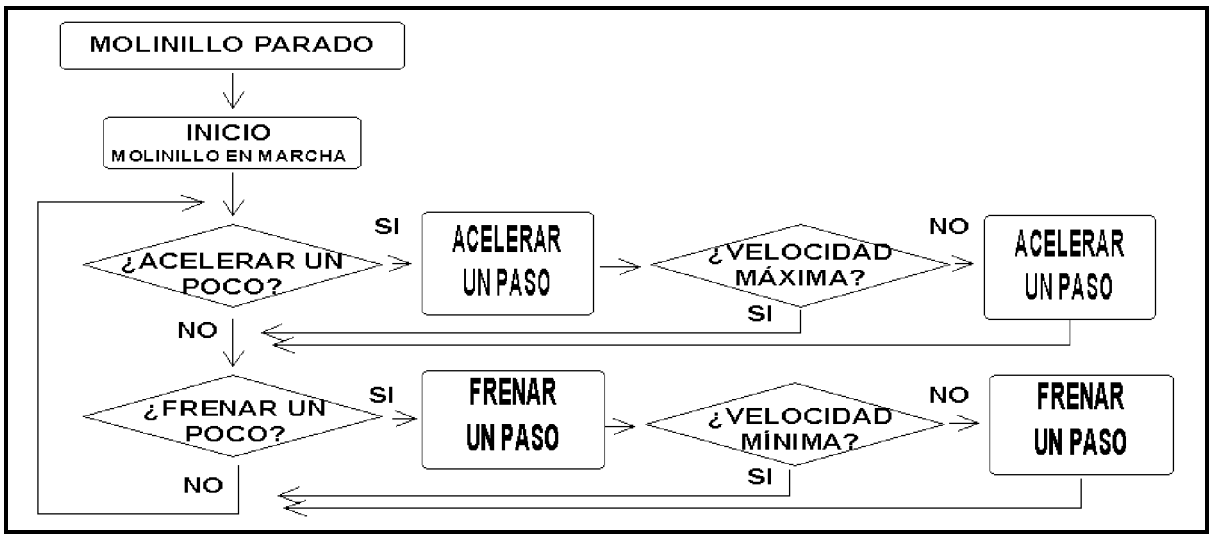
```
PARA ACELERAR_UN_PASO
HAZ "NUM :NUM + 1
SI :NUM > 15 [HAZ "NUM 15 ALTO]
ACELERA :NUM
FIN
```

Este nuevo procedimiento permite aumentar la velocidad paso a paso, hasta que sea igual a 15, impidiendo pasar de ella y evitando de esa manera el error mostrado anteriormente. De igual forma, debe variarse el procedimiento FRENA_UN_PASO para evitar errores.

```
PARA FRENAR_UN_PASO
HAZ "NUM :NUM - 1
SI :NUM < 0 [HAZ "NUM 0 ALTO]
ACELERA :NUM
FIN
```

2.3.A.- MEJORA DE LOS PROCEDIMIENTOS

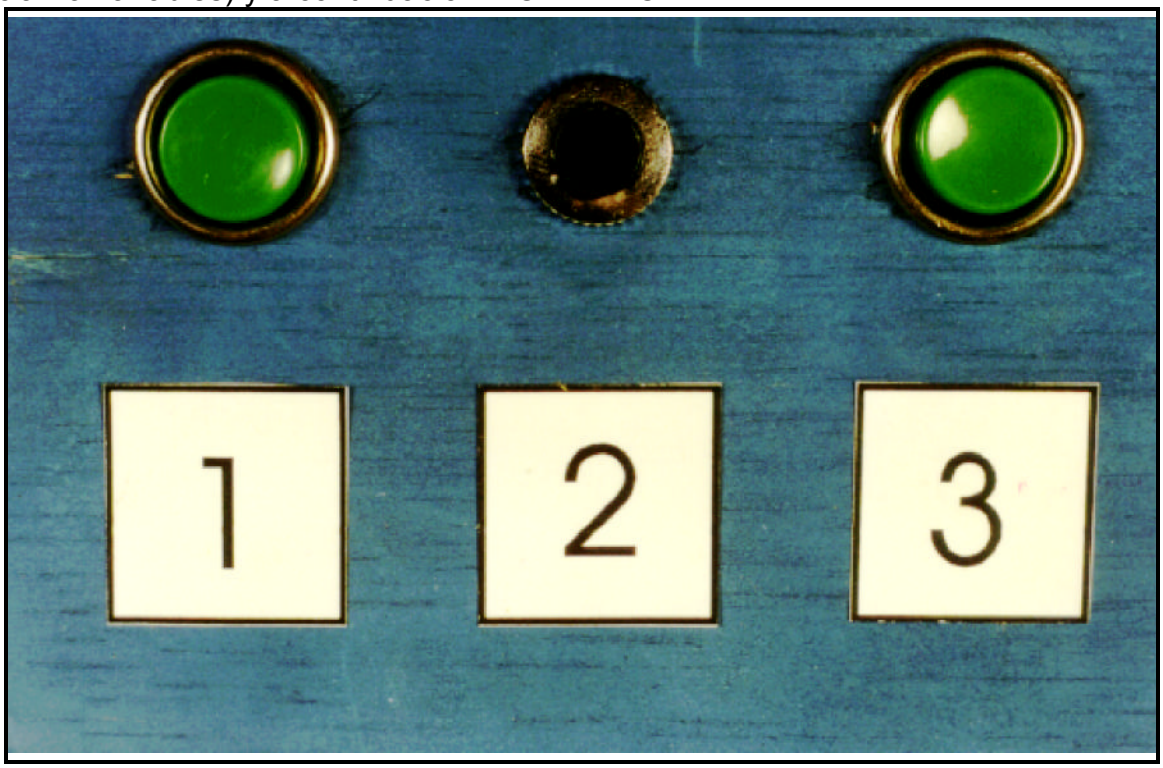
Se han desarrollado tres subprocedimientos INICIO, FRENAR_UN_PASO y ACELERAR_UN_PASO que deben integrarse con la lectura de los tres pulsadores conectados como ya indicamos a +5vcc y a las entradas digitales E1, E2 y E3. Para ello, redefinamos el organigrama. Podría quedar como sigue:



Al procedimiento que controle los sensores digitales lo llamaremos MOLINILLO y podría ser como el siguiente:

```
PARA MOLINLLO
SI SD 2 = 1 [ACELERAR_UN_PASO]
SI SD 3 = 1 [FRENAR_UN_PASO]
MOLINILLO
FIN
```

Para ejecutar todo el conjunto se debe iniciar en primer lugar el procedimiento INICIO (inicializa variables) y a continuación MOLINILLO.



Molinillo. Foto de los tres pulsadores

Ejecutados INICIO y MOLINILLO, al pulsar el pulsador 2 el molinillo acelera y al pulsar el número 3 frena. Para abandonar el flujo del programa debe pulsarse la tecla [Esc] (escape), pues no se ha previsto hasta el momento ningún sistema de salida. Al pulsar [Esc] puede suceder que el motor del molinillo siga en funcionamiento; en tal caso, se debe parar mediante M5 "P escrito desde el área de trabajo.

Una modificación del procedimiento MOLINILLO, que permite abandonar el programa puede ser la siguiente:

```
PARA MOLINILLO
SI SD 2 = 1 [ACELERAR_UN_PASO]
SI SD 3 = 1 [FRENAR_UN_PASO]
SI TECLA? [M5 "P ALTO]
MOLINILLO
FIN
```

Cada vez que se necesita poner en funcionamiento el sistema deben inicializarse las variables ejecutando INICIO; como se tienen tres pulsadores podría usarse uno de

ellos para esta inicialización y, para ello, modificar el procedimiento MOLINILLO, que queda de esta forma:

```
PARA MOLINLLO
SI SD 1 = 1 [INICIO]
SI SD 2 = 1 [ACELERAR_UN_PASO]
SI SD 3 = 1 [FRENAR_UN_PASO]
SI TECLA? [M5 "P ALTO]
MOLINILLO
FIN
```

A partir de este momento, la ejecución de estos programas se realiza escribiendo MOLINILLO y pulsando [Intro].

Para una mayor claridad, se presentan a continuación todos los procedimientos definitivos.

```
PARA MOLINLLO
SI SD 1 = 1 [INICIO]
SI SD 2 = 1 [ACELERAR_UN_PASO]
SI SD 3 = 1 [FRENAR_UN_PASO]
SI TECLA? [M5 "P ALTO]
MOLINILLO
FIN

PARA FRENAR_UN_PASO
HAZ "NUM :NUM - 1
SI :NUM < 0 [HAZ "NUM 0 ALTO]
ACELERA :NUM
FIN

PARA ACELERAR_UN_PASO
HAZ "NUM :NUM + 1
SI :NUM > 15 [HAZ "NUM 15 ALTO]
ACELERA :NUM
FIN

PARA INICIO
HAZ "NUM 0
M5 "D
ACELERA 0
FIN
```

ACTIVIDAD 1

Control de una vagoneta automática.

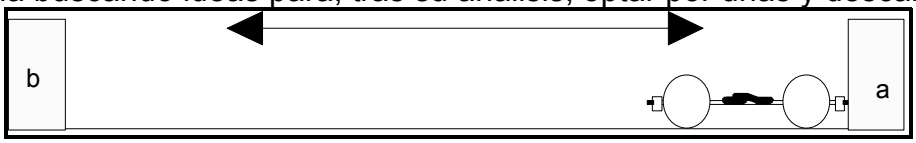
Como en unidades anteriores, aquí también se presenta una propuesta de trabajo diferente a la analizada en este apartado y mostrada en el vídeo para que pueda servir de ayuda y refuerzo a lo ya estudiado. También puede servir como propuesta de trabajo para el aula, teniendo ya el profesor pistas de actuación.

Uno de los objetivos a alcanzar en los primeros años de la enseñanza secundaria es el que el alumnado sea capaz de anticipar el diseño de las soluciones antes de su construcción. En el caso de la robotización, la construcción se refiere a la materialización de la máquina y a la creación de los procedimientos informáticos que permiten su control.

Para que la anticipación sea posible es necesario presentar problemas sencillos y comprensibles para el alumnado, cuya puesta en práctica sea relativamente fácil. El objetivo, como ya se ha indicado, no consiste en construir sino en anticipar. El esfuerzo debe hacerse en los pasos previos a la construcción; como ejemplo de este objetivo veremos una posible construcción y sus pasos previos.

3.1.- PLANTEAMIENTO DEL PROBLEMA

En determinadas actividades industriales deben trasladarse cargas (objetos, paquetes, etc.) no demasiado pesadas entre dos puntos relativamente próximos. Lo normal es utilizar cintas transportadoras. Pero pretendemos variar el concepto de cinta transportadora, y diseñar una vagoneta transportadora, de tal forma que si se varía la distancia por recorrer no sea necesario hacer ninguna trasformación. Hay que pensar el problema buscando ideas para, tras su análisis, optar por unas y descartar otras.

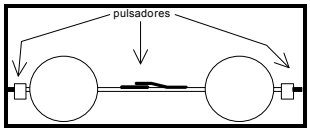


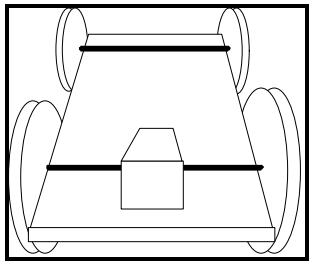
La vagoneta debe desplazarse entre la zona a y la b. En la zona a debe esperar hasta que se coloque una carga sobre ella, momento en el que iniciará su desplazamiento

hacia b. Permaneciendo allí hasta que la descarguen; una vez descargada, debe dirigirse hacia la zona a, detenerse y esperar ser cargada para iniciar un nuevo ciclo.

La vagoneta debe tener un motor que le permita desplazarse, los mecanismos adecuados para detectar su llegada al mostrador a y al b y los que le permitan saber cuando es cargada o descargada. Es adecuado elegir pulsadores como sensores.

La vagoneta podría tener la siguiente estructura:





Un motor con reductora está conectado a un eje de la vagoneta; en la parte anterior y posterior del móvil, a modo de paragolpes, hay dos pulsadores y en la plataforma de carga otro. Los de los extremos sirven para detectar las llegadas a los puntos a y b y el de la plataforma para comprobar si ha sido cargada o está descargada.

PRIMEROS PASOS.

Antes de plantear los posibles pasos a dar para la resolución de problema introduciremos una primitiva de Logo, que puede ser de gran ayuda en este y en futuros planteamientos. La nueva primitiva es MIENTRAS.

Sintaxis: MIENTRAS [Predicado] [Órdenes]

Su funcionamiento es el siguiente: el conjunto de órdenes escritas entre corchetes se ejecutan mientras el predicado escrito también entre corchetes sea cierto.

Ejemplos:

En modo directo podemos escribir

```
MIENTRAS [RUMBO < 90] [ ESCRIBE RUMBO AV 10 GD 90]
```
En el editor podemos probar el siguiente procedimiento.

```
PARA H
HAZ "A 0
MIENTRAS [:A< 10] [AV 10 GD 10 HAZ "A :A +1]
FIN
```

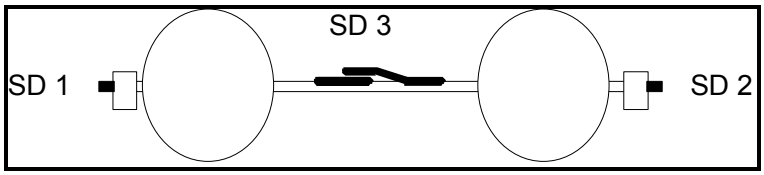
Con estos dos ejemplos puede quedar bastante claro el comportamiento de la primitiva MIENTRAS.

Antes de abordar el proceso de resolución del programa de control, ponte en comunicación con la tutoría si crees no haber comprendido los contenidos de este apartado.

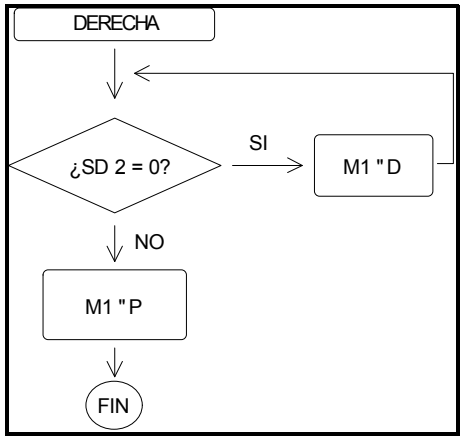
SOLUCIÓN

Volviendo al problema de la vagoneta y pensando en resolverlo mediante pequeños procedimientos nos planteamos como los más importantes el de ir hacia la zona a y el de ir hacia la b. Llamaremos IZQUIERDA el desplazamiento hacia b y DERECHA al desplazamiento hacia a. Para ir hacia la derecha el motor debe estar en marcha hasta que el pulsador de la parte delantera se cierre (SD 2).

Lo mismo para ir hacia la izquierda, aunque en este caso debe controlarse el SD 1.

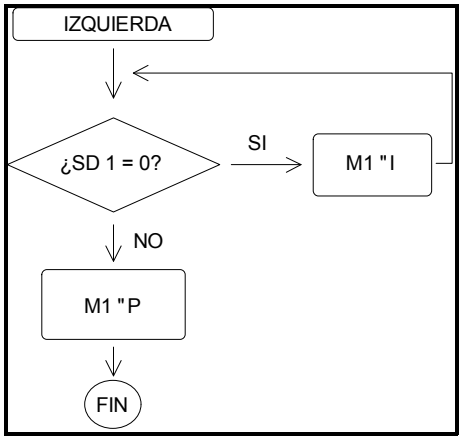


Planteemos el organigrama y el procedimiento de desplazamiento hacia la derecha. Aprovecharemos la primitiva MIENTRAS y programaremos un procedimiento para controlar el desplazamiento de la vagoneta hacia la derecha:




```
PARA DERECHA
MIENTRAS [SD 2 = 0] [M1 "D]
M1 "P
FIN
```

Es conveniente que cada procedimiento, una vez diseñado sea probado realmente.

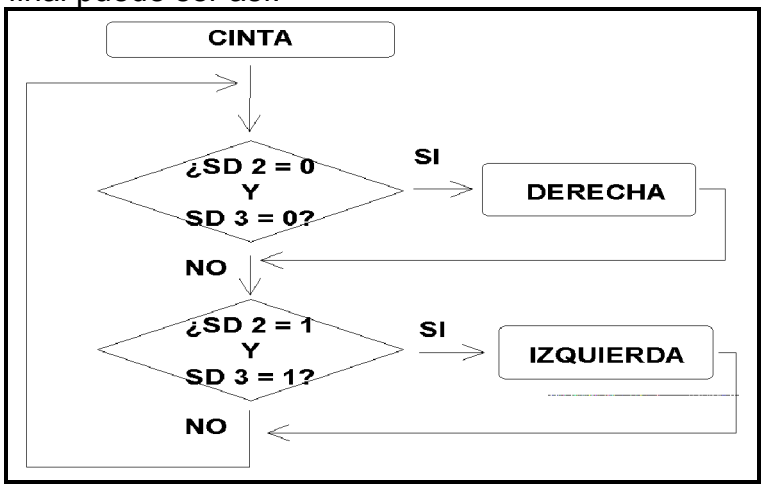


El procedimiento para controlar el desplazamiento hacia la izquierda es idéntico al anterior, aunque cambiamos el sensor por analizar.

PARA IZQUIERDA
MIENTRAS [SD 1 = 0] [M1 "I]
M1 "P
FIN

En ambos procedimientos, tras la salida de la orden MIENTRAS, se ha puesto la orden para parar el motor pues, de lo contrario, seguiría funcionando. Ello se debe a que MIENTRAS no ejecuta ordenes contrarias, sino que detiene el flujo del programa ejecutando la acción ordenada mientras sea cierta la condición expresada.

El organigrama final puede ser así:



Los procedimientos obtenidos desde el organigrama son estos

PARA CINTA
SI Y SD 2 = 0 SD 3 = 0 [DERECHA]
SI Y SD 2 = 1 SD 3 = 1 [IZQUIERDA]
CINTA
FIN

PARA IZQUIERDA
MIENTRAS [SD 1 = 0] [M1 "I]

```
M1 "P
FIN
```

```
PARA DERECHA
MIENTRAS [SD 2 = 0] [M1 "D]
M1 "P
FIN
```

RECUERDA

MIENTRAS [CONDICIÓN] [ACCIÓN]

Mientras la condición sea cierta se ejecuta la acción.

IV.- ENTRE MÁQUINAS Y HERRAMIENTAS

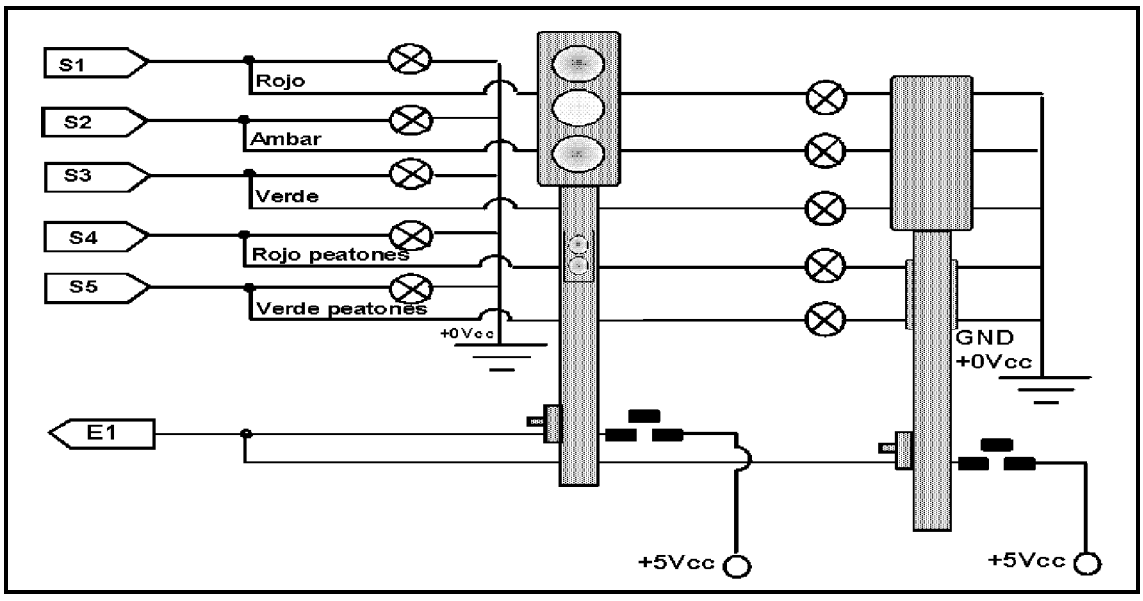
IV.- ENTRE MÁQUINAS Y HERRAMIENTAS

1.- SEMÁFORO CON PULSADOR

> En esta misma sección, en el capítulo anterior, proponíamos el siguiente ejercicio, que vamos a desarrollar ahora. Revisar allí la orden SALIDA VALORBIN "S8_S7_S6_S5_S4_S3_S2_S1_

Colocar los semáforos que permitan el paso de peatones, cada cierto tiempo, en el centro de la longitud de una calle. Para el control, deberá de tenerse en cuenta, además, que los peatones pueden accionar en cada acera el pulsador que les dará prioridad (al accionarlo iniciará el ámbar para los automóviles).

1.1.- CONEXIONES



Las luces rojas de los semáforos de los vehículos se encienden y apagan al mismo tiempo. Por eso se han conectado en paralelos a la salida S1 y a tierra GND.

Otro tanto sucede para las luces amarillas de los semáforos de los vehículos se encienden y apagan al mismo tiempo. Por eso se han conectado en paralelos a la salida S2 y a tierra GND.

Análogamente, las luces verdes de los semáforos de los vehículos se encienden y apagan al mismo tiempo. Por eso se han conectado en paralelos a la salida S3 y a tierra GND.

La luz roja de peatones se ha conectado a S4 y GND. Esta luz deberá coincidir en su conexión con luz verde o ámbar para los automóviles. (Podría ahorrarse está salida digital llevando la línea simultáneamente a S2 y S3, ámbar y verde de los automóviles)

La luz verde de peatones se ha conectado a S5 y GND. Esta luz deberá coincidir en su conexión con luz roja para los automóviles. (Podrían haberse conectado en una misma salida digital)

Basta con que un peatón pulse su botón para que el sistema responda, por ello hemos colocado los dos sensores digitales en paralelo, conectados a la entrada digital E1 y a +5Vcc de nuestra controladora.

1.2.- FASES DEL CICLO

FASE 1 Amarillo para automóviles y rojo para peatones. Supongamos que dura 5 segundos, tiempo necesario para que los vehículos reaccionen frenando. En ese tiempo no se atenderá a la pulsación de los botones ya que la luz verde para peatones viene a continuación.

FASE 2 Rojo para automóviles y verde para peatones. Supongamos que dura 10 segundos, tiempo necesario para que crucen los peatones. En ese tiempo no se atenderá la pulsación de petición de paso ya que los peatones tienen la prioridad.

FASE 3 Verde para los vehículos y rojo para peatones. Supongamos que dura 40 segundos, salvo la petición de los viandantes por pulsación del botón. Tanto si transcurren los 40 segundos como si se pulsa un botón, habrá de reiniciarse el ciclo con la necesaria luz amarilla, para el frenado de los vehículos, que precede a dar el paso a los peatones.

1) La programación de la fase 1 no entraña dificultad. Para activarla las salidas digitales tendrán que ponerse a:

S8 = 0 No influye
S7 = 0 No influye
S6 = 0 No influye
S5 = 0 Apagado verde peatones
S4 = 1 Encendido rojo peatones
S3 = 0 Apagado verde coches
S2 = 1 Encendido ámbar coches

S1 = 0 Apagado rojo coches
La orden que la activa esta fase será:
SALIDA VALORBIN "S8_S7_S6_S5_S4_S3_S2_S1_
SALIDA VALORBIN "00001010

El procedimiento apropiado podría escribirse como sigue:
PARA FASE1
SALIDA VALORBIN "00001010 ESPERA 500 ; Activa la fase y espera 5s.
FIN

2) La programación de la fase tampoco entraña dificultad. Para activarla, las salidas digitales tendrán que ponerse a:
S8 = 0 No influye
S7 = 0 No influye
S6 = 0 No influye
S5 = 1 Encendido verde peatones
S4 = 0 Apagado rojo peatones
S3 = 0 Apagado verde coches
S2 = 0 Apagado ámbar coches
S1 = 1 Encendido rojo coches
La orden que la activa esta fase será:
SALIDA VALORBIN "S8_S7_S6_S5_S4_S3_S2_S1_
SALIDA VALORBIN "00010001

El procedimiento apropiado podría escribirse como sigue:
PARA FASE2
SALIDA VALORBIN "00010001 ESPERA 1000 ; Activa la fase y espera 10s.
FIN

3) La programación de la fase 3 ha de contar con los pulsadores conectados a la entrada del sensor digital SD1. Para activar las luces correspondientes a las salidas digitales oportunas, tendrán que ponerse a:
S8 = 0 No influye
S7 = 0 No influye
S6 = 0 No influye
S5 = 0 Apagado verde peatones
S4 = 1 Encendido rojo peatones
S3 = 1 Encendido verde coches
S2 = 0 Apagado ámbar coches
S1 = 0 Apagado rojo coches

La orden que activa esta fase será:
SALIDA VALORBIN "S8_S7_S6_S5_S4_S3_S2_S1_

SALIDA VALORBIN "00001100

A continuación tendríamos que permanecer el resto de la fase (máximo 40 sg) chequeando el sensor digital SD1 (la E1), con objeto de interrumpirla. Esta operación la desarrollaremos en el procedimiento PULSADO. A expensas de desarrollar este procedimiento el programa apropiado para la fase puede escribirse como sigue:

```
PARA FASE3
SALIDA VALORBIN "00001100 PULSADO ;Activa la fase y pasa a PULSADO.
FIN
```

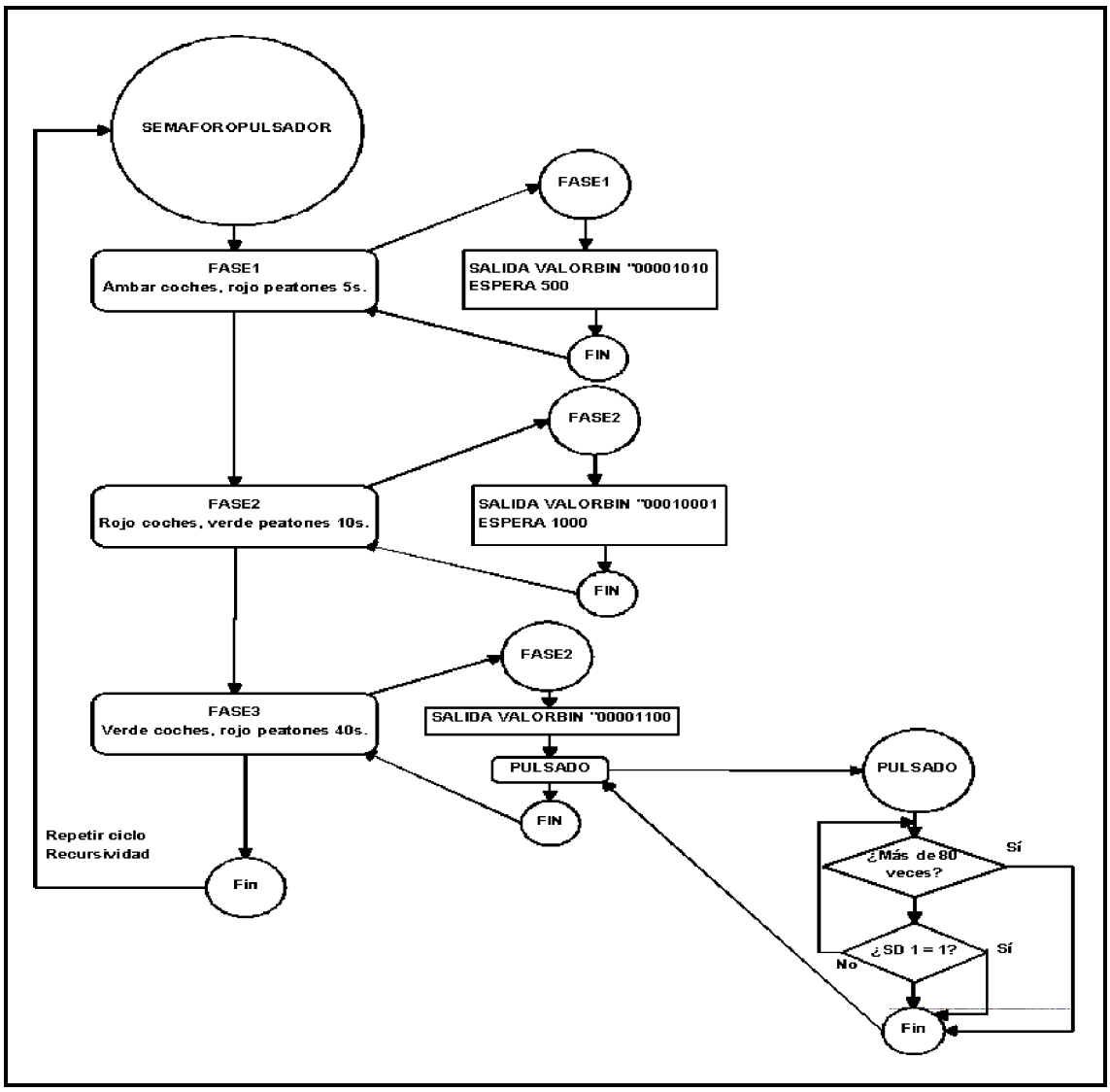
Para desarrollar el procedimiento PULSADO tendremos en cuenta que la controladora invierte un tiempo en chequear el sensor digital. En realidad invierte aproximadamente un segundo para chequearlo dos veces consecutivas. Podríamos mandarle chequearlo 80 veces y dispondríamos de la espera necesaria (40 segundos), no obstante ese número deberá ajustarse experimentalmente. Si en algún chequeo el SD_1 devolviera un "1" (algún pulsador accionado), le ordenaríamos interrumpir el procedimiento PULSADO, con lo que se regresa al procedimiento que lo llamó: FASE3, que al no tener más instrucciones concluiría.

El procedimiento PULSADO podría escribirse así:

```
PARA PULSADO
REPITE 80 [SI SD 1 = 1 [ALTO] ] ;      Repite 80 veces el chequeo, pero si el SD_1
                                       vale "1" se interrumpe (ALTO).
FIN
```

El programa que realizaría cíclicamente (recursivamente) las tres fases sería:

```
PARA SEMAFOROPULSADOR

FASE1;
        Se realiza la fase 1ª (Ámbar para los coches y rojo para peatones,  -5
        segundos- )
FASE2;
        Se realiza la fase 2ª (Rojo para los coches y verde para peatones,  -10
        segundos- )
FASE3;
        Se realiza la fase 3ª (Verde para los coches y rojo para peatones, la fase
        termina si se activa un pulsador o si transcurren 40 segundos )
SEMAFOROPULSADOR;
        Se reinicia el ciclo, llamada recursiva
```


SEMAFOROPULSADOR
FASE1
Ambar coches, rojo peatones 5s.
FASE1
SALIDA VALORBIN "00001010
ESPERA 500
FIN
FASE2
Rojo coches, verde peatones 10s.
FASE2
SALIDA VALORBIN "00010001
ESPERA 1000
FIN
FASE3
Verde coches, rojo peatones 40s.
FASE2
SALIDA VALORBIN "00001100
PULSADO
FIN
PULSADO
¿Más de 80 veces?
Sí
¿SD 1 = 1?
Sí
No
Fin
Repetir ciclo
Recursividad
Fin

FIN

El programa completo quedaría:

```
PARA FASE1
SALIDA VALORBIN "00001010   ESPERA  500
FIN
PARA FASE2
SALIDA VALORBIN "00010001  ESPERA  1000
FIN
PARA PULSADO
REPITE 80 [SI SD 1 = 1 [ALTO] ]
FIN
PARA FASE3
SALIDA VALORBIN "00001100 PULSADO
FIN
PARA SEMAFOROPULSADOR
FASE1
FASE2
FASE3
SEMAFOROPULSADOR
FIN
```

ACTIVIDAD RECOMENDADA

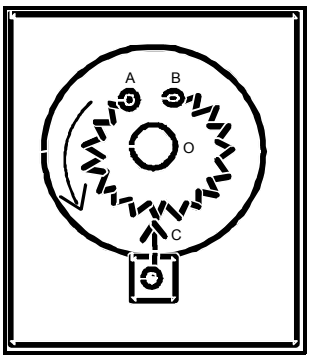
Realizar la construcción del semáforo anterior y hacer las conexiones pertinentes para mejorar el diseño, se pueden ahorrar salidas digitales. Reescribir los programas para esta situación más económica. Conectar el diseño a la controladora. Encender el ordenador, cargar Windows, Win-Logo y el micromundo de control. Editar en la ventana de trabajo el programa realizado, ejecutarlo, afinarlo mediante pruebas y salvarlo.

2.- CONTROL DE UNA GRÚA

Partimos del modelo de grúa construido con los materiales Fischer.

La grúa está equipada con los accesorios de Fischer siguientes:

Dos motores:

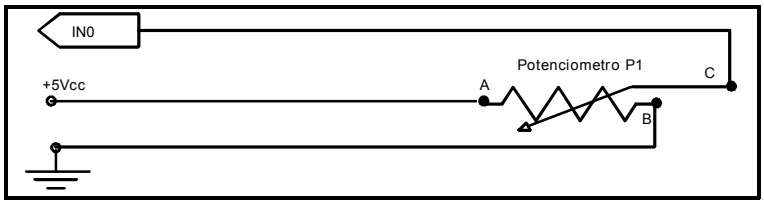


-M1 Que realiza el trabajo de elevación (SUBE) y descenso (BAJA) de las cargas a lo largo del mástil vertical. Este motor se conecta a las salidas digitales S1 y S2.

-M2 Que realiza los giros de la plataforma de sustentación y permite los desplazamientos horizontales de las cargas: El giro hacia la izquierda lo denominaremos IZQUIERDA y el giro hacia la derecha lo llamaremos DERECHA. Este motor se conecta a las salidas digitales S3 y S4.

Dos potenciómetros lineales P1 y P2 de Fischer (4,7 KΩ), esquematizamos uno de ellos en la figura anexa. Estos potenciómetros están sujetos solidariamente a los motores M1 y M2. El giro del motor hace variar la posición del cursor del potenciómetro "C", y por tanto la resistencia R_{AC} es función del desplazamiento (giro) efectuado por el motor. Estos potenciómetros actuarán como sensores analógicos.

El giro del eje del motor, que pasa por "O", obliga a girar a la resistencia "AB". En consecuencia, el cursor "C" determina resistencias variables respecto a sus extremos "A" y "B", que son proporcionales a el giro del motor (desplazamientos del brazo o de la plataforma). La tensión que se aplique en "A" y "B", Va-Vb se repartirá proporcionalmente a los segmentos de resistencia variable: Va-Vc y Vc-Vb. La medición de las variaciones de las correspondientes tensiones Va-Vc, serán medidas proporcionales a los desplazamientos del brazo (o de la plataforma), y la podremos evaluar mediante entradas de sensor analógico: IN0 para el sensor del brazo e IN1 para el sensor de la plataforma.



Las funciones primitivas que leen las entradas analógicas IN0, IN1, IN2 e IN3 de nuestra controladora son: SAX, SAY, SAZ y SAW. La escala de 0 a 5 voltios que pueden leer la dividen en 256 valores, correspondiendo 0 a 0Vcc y 255 a 5Vcc; cada ganancia en un punto en esa escala equivale a 5/256 = 0,0195Vcc.

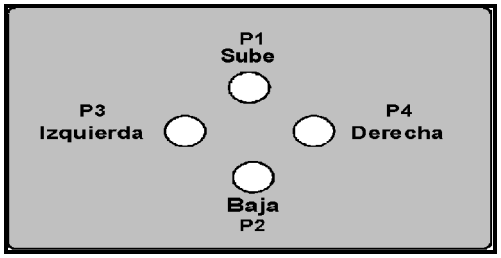
A la hora de trabajar conviene calibrar (determinar experimentalmente) los valores umbrales mínimo (MIN) y máximo (MAX) que nos devuelven los sensores en nuestro montaje, y consecuentemente determinar la correspondencia proporcional entre estos valores numéricos y la variable que mide el sensor.

RECUERDA

SAX,SAY,SAZ y SAW son las funciones primitivas que controlan 4 sensores analógicos conectados a tierra y a las entradas analógicas IN0,IN1,IN2 e IN3, devuelven un valor de 0 a 255 que es proporcional a la diferencia de potencial leída.

Para más información sobre SAX, SAY, SAZ y SAW puedes leer el anexo sobre la controladora.

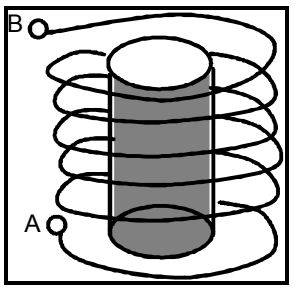
Cuatro pulsadores para controlar los desplazamientos de la grúa



Los 4 pulsadores se instalan sobre un panel de mando para controlar los desplazamientos (SUBE y BAJA) del brazo de la grúa y los giros a IZQUIERDA y DERECHA de la plataforma de la grúa. Serán conectados respectivamente a los sensores digitales SD_1, SD_2, SD_3 y SD_4.

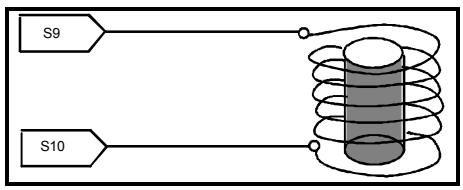
En el apartado "Entre máquinas y herramientas" de la unidad didáctica XIV se utilizó un sistema análogo de pulsadores para controlar los movimientos de un cochecito.

Un electroimán



Se situará el electroimán en el extremo del brazo de la grúa. Al aplicarle distintas diferencias de potencial, Va-Vb, generará en él un campo magnético de intensidad proporcional. Con un potencial apropiado, podremos atrapar por acción de su fuerza magnética diferentes cargas metálicas que liberaremos tan pronto como anulemos dicho potencial.

Para poder aplicar diferentes tensiones al electroimán lo conectaremos a las salidas analógicas de nuestra controladora. Una forma de conexión posible es :



Las salidas digitales S9 y S10 de la controladora pueden activarse con una primitiva Motor, M5, al igual que con M1 se controlaban las salidas digitales S1 y S2.

Así, M5 "D activa la salida S9 con potencial alto y deja la salida S10 con potencial bajo; en cambio M5 "I activa la salida S10 en alta y deja la salida S9 en baja. La

primitiva M5 "P dejará las dos salidas inactivas. Con M5 se puede manejar por lo tanto otro motor, o en nuestro caso el electroimán.

La diferencia de M5 con M1, M2, M3 y M4 es que el nivel de activación de la salidas puede ahora variarse, para ello disponemos de la primitiva ACELERA núm.

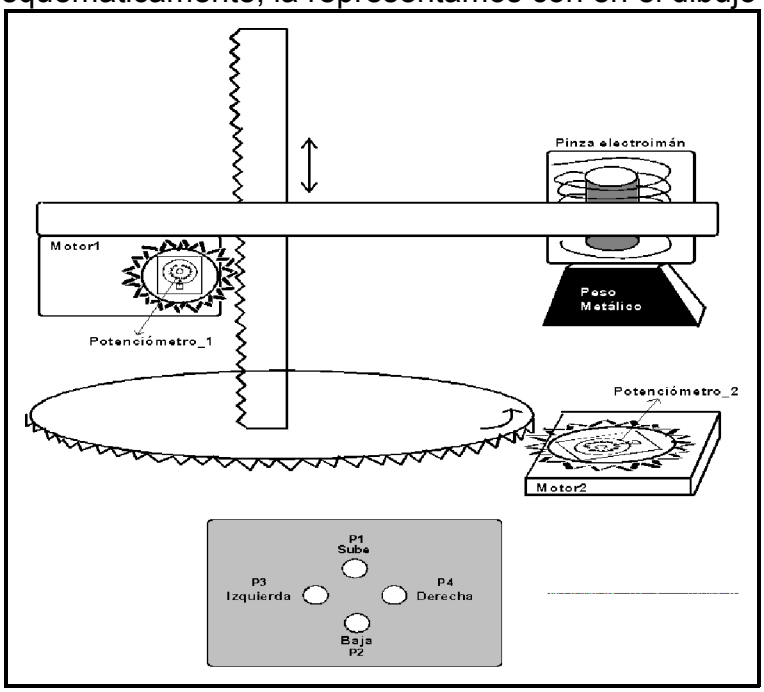
ACELERA núm: Una vez activadas las salidas con M5 "D, ACELERA núm las dejará con una diferencia de voltaje, comprendido entre 2 y 6,5 Vcc., dependiendo linealmente con el valor de núm. El valor núm, valor numérico comprendido entre 0 y 15, distribuirá proporcionalmente ese rango de tensiones, así: si núm = 0 la diferencia de tensión entre S9 y GND (tierra) será 2 Vcc; si núm = 15 (valor máximo) la diferencia de potencial entre S9 y GND será de 6,5 Vcc. Para cada unidad que se incrementa núm, la tensión aumenta en (6,5-2)/15 =0,3 voltios.

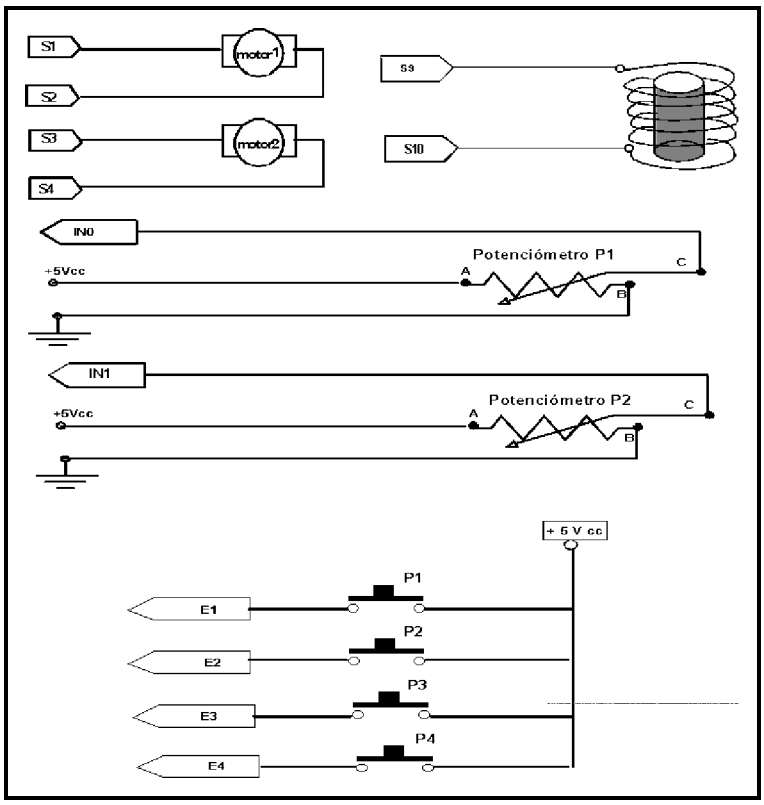
Para más información sobre el control de las salidas analógicas S9 y S10 y las primitivas de control correspondientes, consultar el anexo de este libro sobre la controladora o el manual de la misma.

RECUERDA

Recuerda que M5 "D activa la salida analógica S9, M5 "I la S10 y
M5 "P las desactiva. ACELERA núm (núm de 0 a 15) varía entre 2 y 6'5 V el potencial de activación de las salidas analógicas.

Nuestra grúa, esquemáticamente, la representamos con en el dibujo siguiente:





2.1.- CONEXIONES

Realizar el montaje correspondiente, establecer las conexiones con la controladora, encender el ordenador y la controladora, iniciar Windows, Win-logo y cargar el micromundo de control.

2.2.- CONTROL DE LA GRÚA

1º Comenzaremos calibrando el sensor que controla los desplazamientos verticales sobre el mástil de la grúa (sensor analógico del brazo conectado en IN0).

Si ejecutamos, en modo directo, desde la ventana de trabajo de Win-Logo:

```
M1 "D  ESPERA 10   M1 "P
```

obtendremos un pequeño desplazamiento vertical, convengamos hacia abajo.

Con la orden inversa:

```
M1 "I    ESPERA 10     M1 "P
```

obtendremos un pequeño desplazamiento vertical, en sentido contrario.

Como en lugar de pulsadores, para controlar los fines de carrera, hemos instalado un potenciómetro "P1", como sensor analógico en la entrada IN0, podemos conocer la posición del brazo en cualquier posición mediante:

```
ES   SAX
```

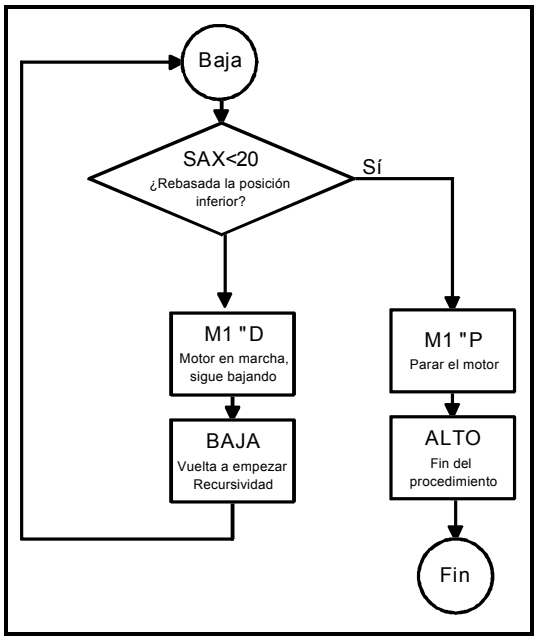
la función SAX devolverá un valor de la altura del brazo que será escrito en la ventana de texto.

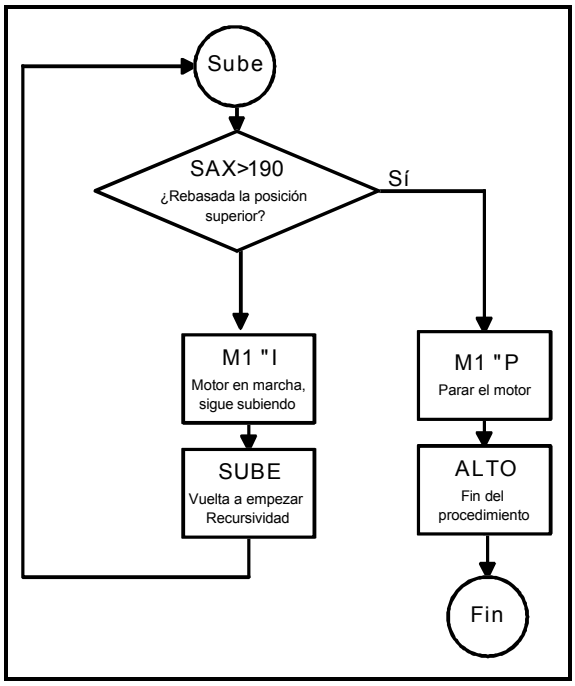
Así, experimentalmente, con pequeños desplazamientos hacia abajo, llevaremos el brazo hasta la altura de recogida y suelta de carga y anotaremos la lectura de esa posición. Supongamos que esa lectura es 20, la denotaremos en cursiva para entender que debe ser substituida por el valor encontrado.

Del mismo modo, con pequeños desplazamientos hacia arriba, llevaremos el brazo hasta la altura de trasporte de la carga y anotaremos la lectura de esa posición.

Supondremos que esa lectura es 190, la denotaremos en cursiva para entender que debe ser substituida por el valor encontrado.

Para obligar a bajar al brazo hasta la posición de recogida de carga podremos escribir el siguiente procedimiento. Análogamente para subir:






```
PARA BAJA
SI   SAX < 20   [M1  "P    ALTO]
M1   "D
BAJA
FIN
```

```
PARA SUBE
SI   SAX > 190   [M1  "P    ALTO]
M1   "I
SUBE
FIN
```

Implementar estos procedimientos y verificarlos, comprobando que los recorridos del brazo de la grúa son los deseados. Del mismo modo procederemos para calibrar los desplazamientos horizontales, giros de la plataforma, y establecer los correspondientes procedimientos.

2º Continuamos, ahora, con el calibrado del sensor que controla los desplazamientos horizontales de la plataforma (sensor analógico de la plataforma conectado en IN1). Los desplazamientos ahora los efectúa el Motor_2.

Si ejecutamos, en modo directo, desde la ventana de trabajo de Win-Logo:

M2 "D ESPERA 10 M2 "P

obtendremos un pequeño giro (desplazamiento horizontal), convengamos hacia la derecha.

Con la orden inversa:

M2 "I ESPERA 10 M2 "P

obtendremos un pequeño giro (desplazamiento horizontal), en sentido contrario.

Como en lugar de pulsadores, para controlar los fines de carrera, hemos instalado el potenciómetro "P2", como sensor analógico en la entrada IN1, podemos conocer la posición horizontal de la carga (giro de la plataforma) en cualquier posición mediante:

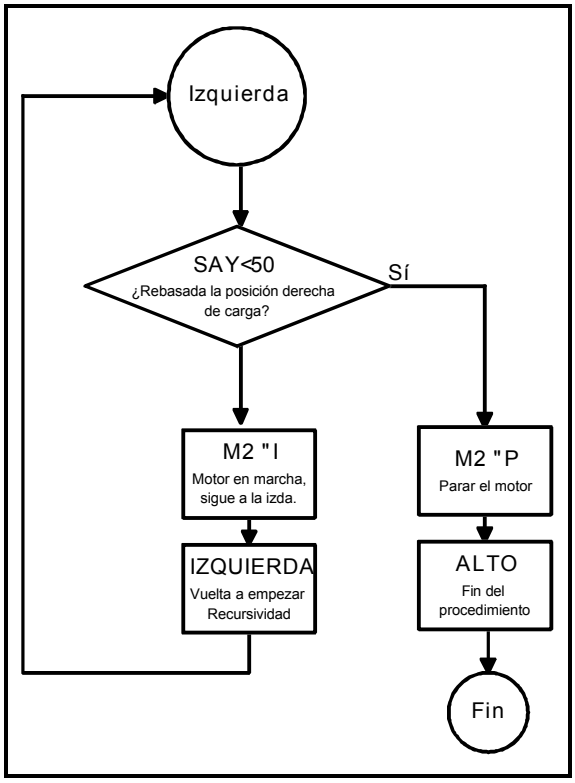
ES SAY

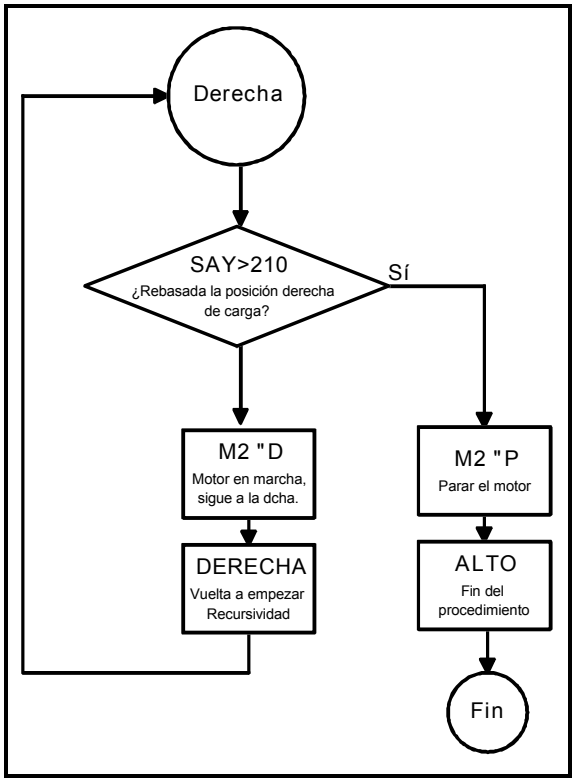
la función SAY devolverá un valor de la posición horizontal, proporcional a la situación angular de la plataforma, que será escrito en la ventana de texto.

Así, experimentalmente, con pequeños desplazamientos hacia la izquierda, orientaremos el brazo, por giro de la plataforma, hasta las posición angular de recogida de la carga, y anotaremos la lectura de esa posición. Supongamos que esa lectura es 50, la denotaremos en cursiva para entender que debe ser substituida por el valor encontrado.

Del mismo modo, con pequeños desplazamientos hacia la derecha, giraremos la plataforma hasta que el brazo esté en la posición angular de descarga y anotaremos esa posición devuelta por SAY. Supondremos que esa lectura es 210, la denotaremos en cursiva para entender que debe ser substituida por el valor encontrado.

Para obligar a girar el brazo, solidario a la plataforma, hasta la posición de recogida de carga (IZQUIERDA) podremos escribir el siguiente procedimiento. Análogamente para girarlo a la posición de descarga (DERECHA):






```
PARA IZQUIERDA
SI  SAY < 50  [M2  "P   ALTO]
M2  "I
IZQUIERDA
FIN
```

```
PARA DERECHA
SI  SAY > 210  [M2  "P   ALTO]
M1  "D
DERECHA
FIN
```

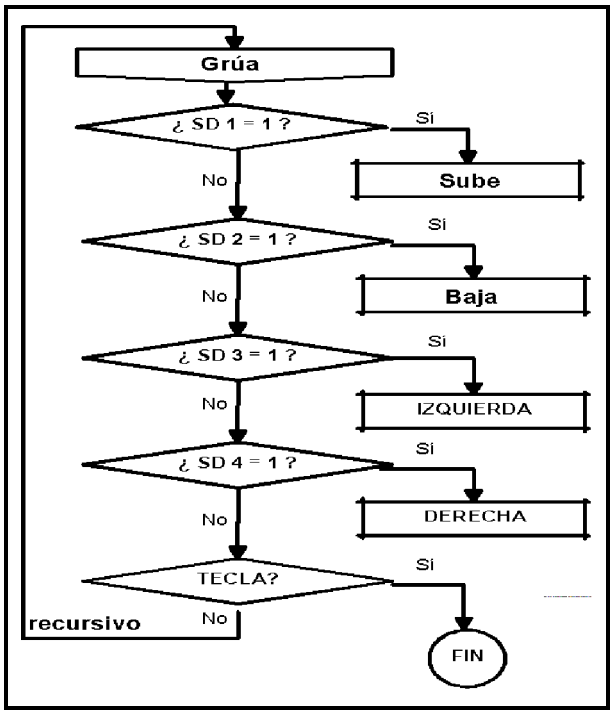
ACTIVIDAD RECOMENDADA

Implementar estos procedimientos y verificarlos, comprobando que los recorridos del brazo de la grúa son los deseados.

3º Veamos ahora el control de estos movimientos mediante los pulsadores.

Esta situación ya se describió con detalle en el apartado " Entre máquinas y herramientas" de la unidad didáctica XIV, al establecer el control de un móvil mediante pulsadores.

Aunque con los procedimientos anteriores ya podríamos efectuar el control de la grúa, mejoraremos éste mediante la utilización de los pulsadores.



Realizamos para ello un procedimiento recursivo que realice el ciclo de retroalimentación.

El procedimiento verifica, uno a uno, cada pulsador ¿ SD n = 1 ?, y si está pulsado, llama al procedimiento de movimiento correspondiente.

Finalmente para salir del bucle recursivo, utilizaremos como condición de parada de la recursividad el condicional SI TECLA? [ALTO], ya tratado en el capítulos anteriores en esta misma sección.

De acuerdo con el organigrama anterior, el procedimiento correspondiente a este control sería:

```
PARA GRUA
SI SD 1 = 1 [SUBE]          ; Pulsado P1, subirá a la posición superior.
SI SD 2 = 1 [BAJA]          ; Pulsado P2, bajará a la posición inferior.
SI SD 3 = 1 [IZQUIERDA]  ; Pulsado P3, girará a izda. hasta su límite.
SI SD 4 = 1 [DERECHA]          ; Pulsado P4, girará a dcha. hasta su límite.
SI TECLA? [ALTO]               ; Pulsando una tecla abandonaremos el control de
                               la grúa, salida de la recursividad.
GRUA                           ; Llamada recursiva. Vuelta a empezar el ciclo
FIN
```

ACTIVIDAD RECOMENDADA

Implementar este procedimiento y verificarlo, comprobando que los movimientos de la grúa al accionar los pulsadores son los apropiados.

4º) Por último, veamos como podemos controlar la pinza, activando y desactivando el electroimán, para atrapar y liberar las cargas metálicas.

Recordemos que nuestro electroimán está conectado a la salidas analógicas: S9 y S10 y que para activarlo, manejado igual que un motor (M5), se utiliza la orden:

M5 "D ACELERA núm

Puesto que nuestro electroimán es débil, le daremos la máxima fuerza (aceleración) posible. Probemos en modo directo: M5 ACELERA 15, y acerquémosle una pieza metálica de tamaño no abusivo, la pieza será sujetada por la pinza.

Para soltarla, bastará desactivar el electroimán, como si de un motor se tratara, ejecutaremos M5 "P y al desactivarse las salidas la pieza metálica se liberará.

Podría suceder que en el núcleo del electroimán, o en la pieza metálica, perdurara un cierto grado de magnetización y la pieza, por muy poco, no se liberaría. Bastaría recubrir con un poco de papel esa pieza y el sistema volvería a funcionar.

Podríamos implementar dos procedimientos SUJETA y SUELTA para conectar y desconectar el electroimán.

```
PARA SUJETA
M5 "D ACELERA 15
FIN

PARA SUELTA
M5 "P
FIN
```

Si queremos realizar un ciclo ininterrumpido, sin control de operario (olvidándonos de los pulsadores), podríamos escribir el siguiente procedimiento:

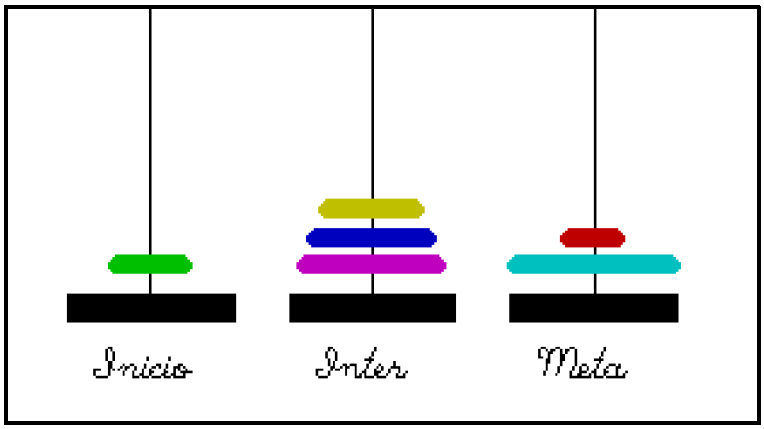
```
PARA GRUA_CICLICA
IZQUIERDA       ;El brazo se desplaza a izquierda (lugar de carga)
BAJA            ;El brazo baja a la posición de carga.
SUJETA          ;La pinza sujeta la carga.
SUBE            ;El brazo sube a la posición de transporte
DERECHA         ;El brazo se desplaza a la derecha (lugar de descarga)
BAJA            ;El brazo baja a la posición de descarga.
SUELTA          ;La pinza suelta la carga.
SUBE            ;El brazo sube a la posición de transporte.
GRUA_CICLICA    ;Llamada recursiva, vuelta a empezar el ciclo.
FIN
```

2.3.- PROPUESTA DE TRABAJO. MEJORA DEL SISTEMA.

Proponemos la inclusión de dos nuevos sensores digitales (pulsadores) para incluirlos en el panel de control, de modo que: al accionar el primero se activara el electroimán y la pinza sujetara las cargas, y al activar el segundo la pinza se desactivara y soltara las cargas.

Otra propuesta sería la reducción de los dos pulsadores a uno sólo que funcionara como variador de estado, cojo_carga-suelto_carga. Podríamos disponer de una bombilla o LED en nuestro panel que nos indicara si la pinza está activada o no.

Una última propuesta, algo más compleja: Habilitar tres posiciones de recogida y descarga de piezas de modo que nuestra grúa resuelva automáticamente el problema de las torres de Hanoi. Puede verse el significado del problema de las torres de Hanoi en uno de los ejemplos que Win-Logo proporciona en con su instalación. Se trata de pasar las fichas desde la posición inicial a la final, sirviéndose de una posición intermedia, pero manteniendo siempre las fichas más pequeñas encima de otras mayores.



V.- CON NUESTROS ALUMNOS Y ALUMNAS

V.- CON NUESTROS ALUMNOS Y ALUMNAS

1.- PROGRAMAS ESTRUCTURADOS

1.1.- PROGRAMACIÓN

Desde el momento que iniciamos a nuestro alumnado en la creación de procedimientos, comenzamos a enseñarles a programar. Fue entonces cuando expusimos lo que podría ser una definición de programación: "Programar es analizar, dividir un problema en pequeños módulos, engarzados entre sí por medio de unas sentencias condicionales". En este tema profundizaremos sobre la sintaxis de estas sentencias

Un programa, estará formado por un conjunto de procedimientos que se van ejecutando según el orden que hayamos establecido y que en un principio sería el mismo que llevamos al leer o escribir en Castellano, de izquierda a derecha y de arriba abajo, a no ser que en cualquier línea del programa se nos mande a otro o, como veremos cuando hablemos de la recursividad, a sí mismo.

Habrá momentos en que el camino a seguir no será único, aparecerán bifurcaciones. En esos casos, el paso de un procedimiento a otro, vendrá determinado por el cumplimiento o no de una serie de alternativas que habremos incluido en unas sentencias condicionales dentro de los propios procedimientos.

Vimos que uno de los puntos claves de los programas está en la capacidad de LOGO de distinguir entre varias opciones en función de unos condicionantes. Éste va a ser el contenido final del tema, pero para poder introducirlo a nuestro alumnado con una base que les permita realizar ejercicios variados y creativos, deberemos previamente mostrar una serie de nuevas primitivas del micromundo de la tortuga: Manejo de variables, Primitivas de control del entorno, mensajes de logo hacia el usuario etc. que vamos a ir presentando previamente a las sentencias condicionales.

Tal como se ha ido viendo en el video, la forma de presentar la estructura de estas primitivas que comienzan con un SI..... será la de que el alumnado asuma que en nuestra vida corriente, estamos tomando decisiones continuamente en función de alternativas y que nuestra elección pasa previamente por un conocimiento de las mismas y unos condicionantes que nos hacen elegir una u otra según se acomoden o no a nuestros deseos previos.

Para dar una idea de uso de las sentencias condicionales que determinan el camino que sigue el desarrollo del programa, podríamos proponer el análisis del problema ACABAR_JORNADA.

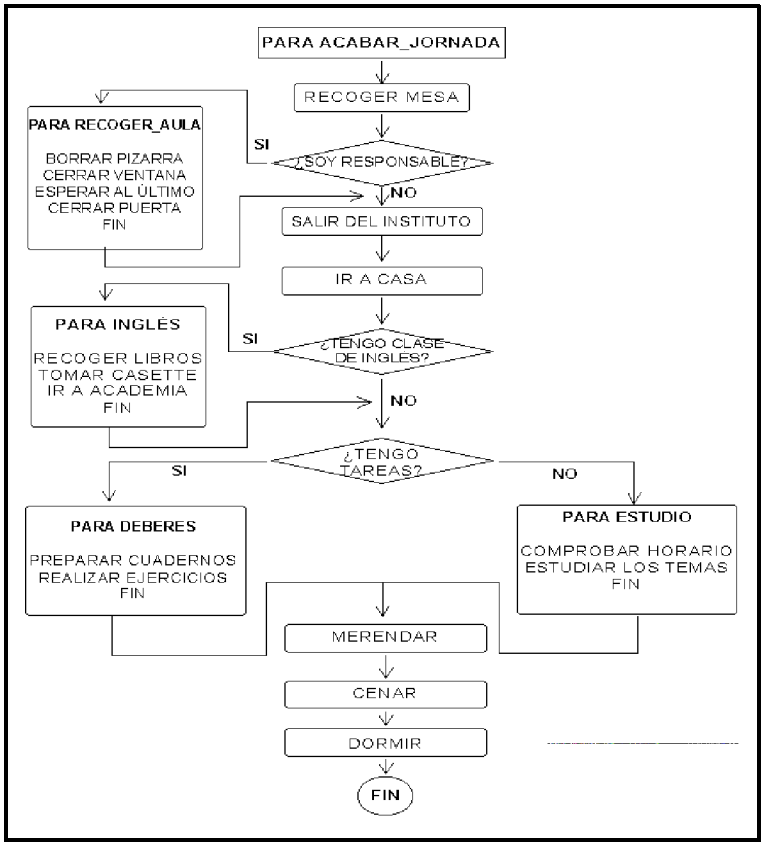
El profesor, previo el trabajo por grupos, deberá dejar claro que el objetivo consiste en reflejar la sucesión de acciones que realiza un alumno desde que suena el timbre de finalización de clase hasta que se acuesta. Dejaremos bien claro que, dependiendo de una serie de condicionantes: día de la semana, responsabilidad de aula etc, las acciones de un día no son iguales a las de otro.

No se pasará a la fase de trabajo de grupo, hasta que todo el alumnado haya comprendido ambos conceptos: Serie de órdenes, sentencias condicionales.

La realización de determinadas tareas requerirá la confección de procedimientos que las desarrollen, procedimientos que serán llamados por el programa principal o iniciador.

En determinados momentos, el programa planteará una opción o seguir el principal y en otros, propondrán elegir entre dos, si no se cumple la primera deberá ejecutarse la segunda, pero siempre volviendo al programa principal que, tal como hemos explicado anteriormente, seguirá el orden natural de nuestra lectura y escritura.

PARA ACABAR_JORNADA
recoger mesa
SI soy responsable [RECOGER_AULA]
salir del instituto
ir a casa
merendar
SI tengo clase de inglés [INGLÉS]
SI tengo tareas [DEBERES] [ESTUDIO]
cenar
dormir
FIN



PARA RECOGER_AULA
borrar pizarra
cerrar ventana
esperar al último
cerrar puerta
FIN

PARA INGLÉS
recoger libro
tomar casette en inglés

ir a academia
FIN

PARA DEBERES
preparar cuadernos
realizar ejercicios
FIN

PARA ESTUDIO
comprobar horario
estudiar los temas
FIN

Será también una buena ocasión para introducir a nuestro alumnado en la representación gráfica del árbol de procedimientos, de cara a que la estructura de los que posteriormente diseñen, sea lo más clara posible y les permita realizar correcciones o mostrarla a otros para que les ayuden a realizarlas.

2.- LAS VARIABLES

Para introducir el concepto de variables, nos apoyaremos en los conocimientos matemáticos que nuestros alumnos y alumnas habrán desarrollado en el área de matemáticas, cuando trabajaron los conceptos de función y ecuación. Sólo tendrán transformar mentalmente la imagen de una x o una y por :l o "r.

2.1.- VARIABLES LOCALES

En el anexo I ya se da una exposición sobre estas variables a las que acompañábamos con un ejemplo utilizando al cuadrado como motivo que nos permitiera la introducción de su sintaxis. De aquella explicación, convendría cuando menos, sacar las connotaciones generales que definen estas variables sobre todo por contraponerlas a las variables globales de las que hablaremos más tarde.

RECUERDA

Las variable locales:

- Se generan en la línea de creación del programa.
- Su declaración va precedida del signo :, por lo que la primera línea tiene la siguiente sintaxis PARA NOMBRE :VARIABLE
- Cuando se quiere hacer referencia a su contenido, debe ir también precedido del signo :
- Su contenido sólo tiene existencia mientras se ejecuta el procedimiento que la genera.
- Otros procedimientos podrían generar variables locales con el mismo nombre.
- Pueden generarse más de una variable por procedimiento, pero todas deben declararse en la primera línea del programa.
- Cuando se va a ejecutar un procedimiento que contiene en su definición una o varias variables locales, el nombre del procedimiento debe ir acompañado del contenido de cada una separándolos por medio de espacios.

2.2.- VARIABLES GLOBALES

Al igual que con las anteriores, las ideas fundamentales están recogidas en el anexo I, posteriormente plantearemos la forma de introducirlas en nuestra sesión de LOGO, ahora destacaremos los aspectos que consideramos mejor las definen en contraposición a las locales.

RECUERDA

Las variables globales:

- Se generan en cualquier línea interior del programa.
- Su declaración se consigue con la primitiva HAZ "nombre seguida del contenido que se le quiere dar
- Cuando se quiere hacer referencia a su contenido, el nombre de la variable debe ir precedido del signo :
- Su contenido tiene existencia mientras se trabaja en área.
- Si se crearan otros procedimientos con variables con el mismo nombre, se interferirían tomando el último contenido asignado.
- La forma de evitar estas interferencias es declararlas previamente y localizarlas por medio de la primitiva LOCAL [VAR1, VAR2,...] seguida de los nombres de las variables que se quieren generar posteriormente. Su contenido tendrá valor dentro del procedimiento que las genera y todos aquellos que lo llamen o sean llamados por él.
- Pueden generarse más de una variable por procedimiento.
- Hay que distinguir claramente la expresión "variable que supone la creación de una variable de :variable que hace referencia al contenido de la variable definida con ese nombre.
- Cuando el contenido de una variable es una palabra, irá precedido de comillas y cuando sea una lista, irá entre corchetes.

La clarificación de estos conceptos debe ser el objetivo de las sesiones dedicadas al tema de las variables, pero para poder sacar provecho de las mismas en cualquier tipo de ejercicios que propongamos o mostremos a nuestro alumnado, deberá ir acompañada de la presentación de otras primitivas que iremos presentando prácticamente a la par.

2.3.- METODOLOGÍA PARA LA PRESENTACIÓN DE LAS VARIABLES LOCALES.

Tal como ya hemos expuesto en otros apartados, la mejor motivación para estos conceptos es la de crear la necesidad de utilizarlos, que sea el propio alumnado quien

las pida. No vamos a pretender que obliguemos al grupo a realizar cuadrados de todas las medidas para que, aburridos de repetir cambiando un número pidan a gritos un sistema más cómo y racional para hacerlo, sobre todo cuando el concepto de variable, ya forma parte de su bagaje matemático.

Convendrá, de todos modos, forzar la necesidad de tener un procedimiento que manteniendo unas características generales, pueda admitir una medida variable que le permita adaptarse más que a una figura, a un concepto.

El trabajo sobre el cuadrado, tal como se explicó en el Anexo I, puede ser llevado ante los alumnos prácticamente siguiendo los mismos pasos. Para presentar una variación en este apartado, proponemos como objeto tipo, a la circunferencia.

1º Creación de un programa que dibuje una circunferencia única (polígono de 36 lados, puesto que para la definición de la pantalla no hay que acercarse más al infinito). (La parte geométrica de CIR). Hacer ver que, para realizar otra circunferencia hay que modificar sólo el avance, pero seguiremos creando una circunferencia concreta.

2º Creación de un procedimiento que permita dibujar distintas circunferencias en función del avance que de la tortuga en cada uno de sus 36 pasos. (La parte geométrica de CIRVAR :D). Plantearemos un símil comentando que es como si la tortuga tuviera un bolsillo con una etiqueta por fuera. En la primera línea es donde se crea el bolsillo dándole un nombre a la etiqueta, en este caso :D. Conforme diseñamos el procedimiento, se puede hacer mención del contenido del bolsillo para que opere con él, ya escribiéndolo, rotulándolo, usándolo como argumento, operando con él, etc, en este caso como argumento de la primitiva AVANZA. Siempre que se hace mención del contenido del bolsillo, se escribe :nombre, aquí :D. Cuando ejecutemos el procedimiento que contiene el bolsillo, debemos darle un contenido, de otro modo advertiría que le faltan datos, como un argumento de algunas primitivas, aquí CIRVAR 12. Cuando la tortuga está ejecutando un procedimiento que contiene una variable, ejecuta las órdenes una tras otra, pero cuando llega a una expresión donde escribimos que avanzara, girara, o escribiera :nombre, mira su bolsillo y escribe, gira o avanza el contenido allí guardado, en este caso AVANZA 12.

3º Ejecutar el procedimiento CIRVAR varias veces cambiando el argumento.

4º Mostrar en modo directo las primitivas ROTULA y ESCRIBE haciendo notar donde se ejecuta cada una de ellas y la necesidad de acompañar sus argumentos con comillas o corchetes según se trate de palabras o frases. Presentar también en modo directo la primitiva FRASE como una operación de suma de dos sumandos.

5º Completar los procedimientos CIR y CIRVAR :D con la rotulación en la pantalla de gráficos.

6º Realización de modificaciones al programa anterior para introducirle más variables como colores, grosores, fondos, además de la longitud ya mencionada. Explicar que la tortuga puede tener varios bolsillos a la vez con diferentes etiquetas, que pueden ser usados del modo expuesto anteriormente. En su ejecución, debemos llenarlos en el orden que los hemos definido. Como ejemplo puede servir CUAVARIA :C :G :F del Anexo I.

7º Diseño de un programa que introduzca la posibilidad de crear cualquier tipo de polígonos teniendo en cuenta el viaje total y la capacidad operativa de LOGO (POLIN :N de Anexo I).

8º Crear un programa que dibuje una circunferencia a partir de un punto de ella y que nos pueda servir de modulo para otros procedimientos, es decir que no empiece marcando las condiciones generales en que deba trabajar, tan solo que dibuje a partir de donde se encuentre y en la posición en que estaba. (CIRLAT :MODUL).

9º Crear procedimientos que contengan variables locales y que realicen formas distintas a base de repetir circunferencias de distintos tamaños y moverse después por medio de avances o giros. (ROSA :M, TUBULAR :T, ESTREL : MODUL).

10º Confeccionar un procedimiento que dibuje una circunferencia en función de su radio, partiendo y dejando a la tortuga en el centro.(CIRRAD :RADIO). Antes de realizar esta propuesta, pondremos a disposición del alumnado la primitiva PI, para lo que usaremos simplemente el modo directo : ES PI.

Tanto en este procedimiento como en CIRLAT, que serán usados como módulos para otros, al alumnado no se le ocurrirá dar el giro de 5º antes y después del REPITE, pues la figura es correcta sin ellos. Su presencia se hace necesaria en otros procedimientos que comentaremos posteriormente y que requieren de ajustes finos con otras figuras.

11º Diseñar procedimientos que requieran la presencia de variables locales y que jueguen con la repetición de circunferencias centradas de distintas dimensiones junto con desplazamientos y giros.(TABA :TUB).

Las aplicaciones de estos programas como desarrollo y profundización de conceptos geométricos se pueden trabajar con propuestas de diseño de procedimientos que realicen circunferencias inscritas o circunscritas a polígonos regulares. Para ello, deberán diseñar primero, procedimientos que dibujen un cuadrado y un hexágono en función de su lado (CUAV :L, EXAV :L).

12º Crear procedimientos con variables locales que, llamando a un cuadrado o un hexágono, los inscriban o circunscriban a una circunferencia. (CUACIR :R, CUAINS :R EXACIR :R, EXAINS :R). Para poder realizar esta propuesta, deberemos presentar, también en modo directo, la primitiva RC: ES RC 16. A pesar de ello, antes de entrar en el diseño de estos procedimientos, nos aseguraremos que los cálculos matemáticos que los posibilitan están claros en la mente de nuestros alumnos y alumnas.

13º Mejorar estos procedimientos cambiando sus colores, coloreando su interior con la primitiva RELLENA, dibujando con colores sus distintos elementos como radios, apotemas etc., o realizando el cálculo de sus superficies.

2.4.- METODOLOGÍA PARA LA PRESENTACIÓN DE LAS VARIABLES GLOBALES

Seguiremos usando el símil que utilizábamos para introducir las variables locales a base de relacionar la creación de variables que tienen un nombre y un contenido con unos bolsillos con etiqueta externa. Las diferencias están en que su creación se realiza en cualquier línea del programa (excepto la primera PARA.. y la última FIN) y que su existencia no se circunscribe a la duración del procedimiento.

Deberemos presentar previamente las primitivas LeeCarácter y LeePalabra, para ello elaboraremos programas sencillos que muestren cómo ambas requieren la actuación del usuario con la diferencia de que la primera actúa con la primer pulsación y la otra necesita la confirmación por medio de la tecla [INTRO]. (PULSA) (INTRO). Aprovecharemos el segundo para explicar que la sintaxis de la primitiva FRASE se simplifica al englobar dentro de un paréntesis tanto a ella como a los textos que manipula y las variables que queramos unir.

Explicaremos que en PULSA diseñamos un programa que con la máxima simplicidad, limpia la pantalla de textos, muestra el mensaje en la pantalla de textos y devuelve en la misma pantalla la pulsación que ha realizado el usuario unida a otro texto por medio de la conocida primitiva FRASE.

Aprovecharemos el programa INTRO, para explicar la utilidad de la primitiva ESPERA, que nos sirve para dar tiempo fijo a la lectura. La diferencia entre LC y LP se hará evidente en la ejecución de uno y otro, por lo que sólo nos quedará insistir en la utilidad del paréntesis al usar la primitiva FRASE.

En el primer programa (CIRGLO1), propondremos crear un procedimiento que dibuje una circunferencia de radio variable, pero que, a diferencia de CIRVAR, indique al usuario los pasos que debe seguir por medio de mensajes en la pantalla de textos. Una vez introducido el dato requerido, el programa lo introducirá en una variable y actuará como en CIRVAR.

El siguiente paso será conseguir que el programa realice la circunferencia variable y centrada en función del radio, como CIRRAD, pero realizando y presentando el cálculo de la longitud de la circunferencia. (CIRCEN1).

Como rápidamente apreciarán el parecido de las líneas de actuación de la tortuga con las que tienen CIRLAT y CIRRAD, las siguientes propuestas de ejercicios consistirán en realizar procedimientos que, llamando a los procedimientos anteriores, efectúen las acciones que estos realizaban con la mejora que supone el que, por medio de mensajes que el programa envía al usuario, éste va siguiendo las instrucciones y, sin necesidad de conocer el programa, lo pone en marcha dándole los datos necesarios para su ejecución. (CIRGLOBAL CIRCENTRAL). Así mismo, servirán para manejar las nuevas primitivas LC y LP y las distintas formas de comunicación entre LOGO y el usuario ESCRIBE y ROTULA.

Mostraremos cómo el contenido de las variables sigue existiendo después de la ejecución de estos programas con órdenes directas como Escribe :mod o Rotula :CIRCUM.

La presentación de la primitiva ENTERO, puede hacerse por medio de un sencillo programa diseñado con una variable local que combine LP y esta primitiva. quite decimales. (DECENTE). Su estructura es similar a PULSA o INTRO, con mensajes en la pantalla de texto, tan solo presenta como novedad la inclusión de la primitiva LC como forma de control del flujo de pantallas discrecional para el usuario. El uso que se le da como"bolsillo" para una variable es posterior

3.- SENTENCIAS CONDICIONALES

3.1.- METODOLOGÍA

Retomamos la definición expresada al principio de este tema: Programar es analizar, dividir un problema en pequeños módulos, engarzados entre sí por medio de unas sentencias condicionales.

Nuestro alumnado, ha ido realizando programas que se ejecutaban de forma lineal, en el orden de la escritura occidental; los programas se engarzaban entre sí llamándose unos a otros. En este apartado presentaremos la posibilidad que nos ofrece LOGO de crear bifurcaciones, distintos caminos que serán recorridos opcionalmente en función de sentencias condicionales.

No insistiremos en la explicación de la sintaxis de la primitiva SI pues ya fue expuesta en el Anexo I. Simplemente recordaremos el funcionamiento interno de sus dos opciones.

Se puede expresar SI Condición [Acción].

Esta sintaxis presenta dos caminos, realizar la acción o no realizarla; Expresado de otro modo, tomar un desvío o no tomarlo. En ambos casos, seguirá con la ejecución de la línea principal del programa, retomará el camino principal si se ha desviado.
La otra expresión será SI Condición [Acción1] [Acción2]

En esta ocasión, obligará a tomar un desvío u otro en función de cumplirse o no la condición expuesta, y en ambos casos, volverá al programa principal una vez ejecutadas una u otra acción.

Para presentar estas sentencias con un ejemplo, nos apoyaremos en el gráfico que hicimos al principio de este capítulo en el que estructurábamos el programa ACABAR_JORNADA.

Insistiremos en la diferencia que existe entre los condicionales:
SI soy responsable o SI tengo clase de inglés; puesto que en ambos casos o se toma el desvío o se sigue el camino principal.
Con el condicional SI tengo tareas, que nos lleva a un desvío o a otro para acabar en ambos casos en el programa principal.

En estos primeros pasos podemos presentar sencillos programas para que el alumnado realice los esquemas y comente cuál será el posible flujo en función de las posibles respuestas dadas, antes de escribirlo en el ordenador.

Otro ejemplo que podemos diseñar para mostrar estas diferencias sería el dado por el programa MACHISMO. En un primer momento, usaremos sólo el condicional SI LC="o, pero al comprobar que las letras mayúsculas provocan un efecto "femenino", aprovecharemos para introducir la sintaxis de la expresión lógica O (MACHISMOO), que permitirá la entrada de la O mayúscula como válida.

3.2.- LOS FILTROS

Una de las utilidades prácticas de las sentencias condicionales es la de la elaboración de filtros que impidan las interrupciones que pueden producirse en el flujo de los programas por la introducción por parte del usuario de datos no idóneos. La motivación que tendremos para presentarlos a nuestros alumnos y alumnas deberá ser, la necesidad de proteger nuestros programas de los errores de los posibles usuarios de los mismos.

Partiremos de programas ya diseñados como CIRGLO Y CIRCEN o los que hemos creado para presentar algunas primitivas como PULSA. Empezaremos ejecutándolos e introduciendo en ellos datos no idóneos:
En CIRGLO a la pregunta "Escribe la cifra del avance", escribiremos la letra O.
En CIRCEN para escribir la longitud del radio escribiremos DIEZ y pulsaremos intro.

Al observar la ruptura del programa y el mensaje de error que nos comunica, plantearemos a nuestro alumnado la necesidad de contar con un filtro que, discriminando la entrada dada por el usuario, compruebe si es o no idónea y en este caso, desvíe el flujo del programa hacia el inicio para evitar su interrupción.

Presentaremos la primitiva NUMERO? obj como condición a introducir en una sentencia y que al devolver Cierto o Falso nos permitirá retroceder o seguir según le indiquemos a continuación entre corchetes.

```
PARA FILNUM :MOD
SI NO NUMERO? :MOD [ES [Este dato no es un número, escríbelo correctamente]
CIRGLO]
FIN
```

Ya solo necesitamos modificar el programa CIRGLO colocando este filtro después de la creación de la variable "MOD

```
PARA CIRGLO
.....
.....
HAZ "MOD LC
FILNUM :MOD
CIRVAR :MOD
....
FIN
```

El mismo filtro nos servirá para proteger a CIRCEN con sólo modificar el nombre del programa al que deberá volver si se comete un error en la entrada. De todos modos, aunque como solución del problema planteado es suficiente, se aprecia que el retorno, en este caso, adolece del defecto de volver al inicio y no al lugar donde se cometió el fallo, con lo que lo primero que ejecutará el programa en su retorno será un BT que impedirá la lectura. Ni que decir tiene que un ESPERA antes de CIRGLO parcheará el problema, pero debemos aclarar entonces que si queremos mejorar el programa, es preferible olvidar la estructura anterior dándole una estructura modular nueva.

3.3.- LOS MENÚS

Otra aplicación de las sentencias condicionales la tenemos en la posibilidad que ofrece al programador para presentar menús con varias opciones para que el usuario pueda optar entre ellas.

Intentaremos presentarlas por medio de una aplicación inmediata a los programas creados en este mismo tema. Propondremos que sirva como pantalla de presentación y menú de elección algunos de los programas desarrollados en este tema: CIRGLOBAL, CIRCENTRAL, y CIR. Posteriormente, introduciremos modificaciones

para que se puedan introducir también,aquellos programas que se confeccionaron con variables locales y propusimos fueran llamados por otros que después de crear una variable global, llamaban a estos para ejecutarlos. CUAINS, EXAIN, CUACIR, EXACIR, etc. El programa base (CIRCUMLOQUIO), lo daremos confeccionado para que lo completen y mejoren pudiendo además, servirnos de engarce con los recursivos.

- Completar el menú incluyendo otros programas diseñados anteriormente con variables globales y mensajes en la pantalla que dirigen su ejecución.

ANEXO AL TEMA XVII

```
PARA CIR
OT BP BL PONCL 1 PONG 1 PONF 16
REPITE 36 [AV 10 GD 10]
SL RE 70 GD 90 RE 100 BL
ROTULA "Circunferencia
SL CENTRO MT
FIN

PARA CIRCEN1
BP BT
ES [Este programa dibuja una circunferencia en función del radio que tu elijas]
ESPERA 100
ES [Escribe con cifras la longitud del radio]
ES [PULSA INTRO]
HAZ "RAD LP
BL AV :RAD GD 90
REPITE 36 [AV 2*PI*:RAD/36 GD 10]
SL GI 90 CENTRO
FIN

PARA CIRCENTRAL
BP BT
ES [Este programa dibuja una circunferencia en función del radio que tu elijas]
ESPERA 100
ES [Escribe con cifras la longitud del radio]
ES [PULSA INTRO]
HAZ "RAD LP
BL AV :RAD GD 90
HAZ "CIRCUM 2 * PI * :RAD
REPITE 36 [AV :CIRCUM / 36 GD 10]
SL MT RE 200
BL ROTULA ( FR [El radio mide] :RAD [pasos] )
BT ES ( FR [La circunferencia mide] ENTERO :CIRCUM [pasos] )
SL CENTRO
FIN
```

```
PARA CIRGLO1
OT BP BL PONCL 1 PONG 1 PONF 16
BT ESCRIBE [Escribe la cifra de avance de la tortuga para que dibuje una circunferencia]
HAZ "MOD LC
REPITE 36 [AV :MOD GD 10]
SL CENTRO
FIN

PARA CIRGLOBAL
OT BP BL PONCL 1 PONG 1 PONF 16
BT ESCRIBE [Escribe la cifra de avance de la tortuga para que dibuje una circunferencia]
HAZ "MOD LC
CIRVAR :MOD
SL CENTRO
FIN

PARA CIRVAR :D
OT BP BL PONCL 1 PONG 1 PONF 16
REPITE 36 [AV :D GD 10]
MT SL RE :D * 7 GD 90 RE 100 BL
ROTULA FRASE [circunferencia de] :D * 36
FIN

PARA CIRLAT :MODUL
BL OT
GD 5 REPITE 36 [AV :MODUL GD 10] GI 5
MT
FIN

PARA CIRRAD :RADIO
SL AV :RADIO GD 90 BL OT
GD 5 REPITE 36 [AV 2 * PI * :RADIO / 36 GD 10] GI 5
SL GI 90 RE :RADIO MT
FIN

PARA CUAV :L
REPITE 4 [AV :L GD 90]
FIN
```

```
PARA CUACIR :R
BP PONCL 1 PONG 1
PONF 16
CIRRAD :R
SL RE :R GI 90 RE :R BL
CUAV :R * 2
SL CENTRO
FIN

PARA CUAINS :R
BP PONCL 1 PONG 1 PONF 16
CIRRAD :R
SL GI 45 RE :R GI 45 BL
CUAV :R * RC 2
SL CENTRO
FIN

PARA DECENTE
BT ES [Este programa suprime los decimales de los números que escribas]
ES []
ES [Pulsa una tecla para continuar]
HAZ "Q LC
BT
ES [Escribe un número decimal y pulsa después intro]
HAZ "Q LP
ES [La parte entera del decimal que has escrito es]
ES ENTERO :Q
FIN

PARA ESTREL :MODUL
BP BL PONCL 1 PONG 1 PONF 16 GI 90
REPITE 5 [ROSA 1 SL AV 50 BL GD 360 / 5] GD 90
FIN

PARA EXAV :L
REPITE 6 [AV :L GD 60]
FIN
```

```
PARA EXACIR :R
BP PONCL 1 PONG 1
PONF 16
CIRRAD :R
SL RE :R GI 90 RE ( :R * RC 3 ) / 3 BL
EXAV ( 2 * :R * RC 3 ) / 3
SL CENTRO
FIN

PARA EXAINS :R
BP PONCL 1 PONG 1 PONF 16
CIRRAD :R
SL GI 30 RE :R GI 60 BL
EXAV :R
SL CENTRO
FIN

PARA ROSA :M
REPITE 9 [CIRLAT :M CIRLAT :M + 1 GD 360 / 9]
FIN

PARA INTRO
BT ES [Para contestar a este programa]
ES [debes pulsar intro después de cara respuesta]
ESPERA 300 BT
ES [¿Cuántos años tienes?]
ES (FR [El año que viene cumplirás] LP+1 [años])
FIN

PARA MACHISMO
BT ES [Si eres chico pulsa una o]
ES [Si eres chica pulsa otra tecla a]
SI  LC ="o [Así que te pareces a tarzán?] [Así que te pareces a Eva?]
ESPERA 200
ES [¿Cuantos años tienes?]
SI LC<30 [Eres una persona joven]
ESPERA 200
ES [Vive intensamente]
FIN
```

```
PARA MACHISMOO
BT ES [si eres chico pulsa la letra o]
ES [si eres chica pulsa otra tecla]
HAZ "BOLSO LC
SI O :BOLSO ="O :BOLSO="O [ES[Así que te pareces a Tarzán?]] [ES[Así que te pareces a Eva?]
ESPERA 200
ES [¿Cuántos años tienes?]
HAZ "BOLSO LP
SI :BOLSO<30 [ES [Eres una persona joven]]
ESPERA 200
ES[Vive intensamente]
FIN

PARA PULSA
BT
ES [Pulsa la letra inicial de tu nombre]
ES FRASE [Tu nombre empieza por] :LC]
FIN

PARA TUBULAR :T
BP BL PONCL 1 PONG 1 PONF 16
REPITE 15 [OT CIRLAT :T SL GD 360 / 30 BL AV 15 CIRLAT :T + 2 MT]
FIN

PARA TUBA :TUB
BP PONCL 1 PONG 1 PONF 16
SL RE 50
REPITE 20 [CIRRAD :TUB CIRRAD :TUB + 5 AV 5]
FIN

PARA CIRCUMLOQUIO
PANTALLA
MENU
FIN

PARA PANTALLA
BP BT PONF 16 GD 90 SL RE 50 BL
ROTULA "CIRCUMLOQUIO
ESPERA 400 BP
SL GD 90 RE 250 BL
ROTULA [1 CIRCENTRAL - 2 CIRGLOBAL - 3 CIR]
ES [Elige el número de la opción deseada]
```

FIN

```
PARA MENU
HAZ "ORDEN LC
SI NO NÚMERO? :ORDEN [ES [Escribe un número con cifras] MENU]
SI :ORDEN 1 [CIRCEN]
SI :ORDEN 2 [CIRGLO]
SI :ORDEN 3 [CIR
FIN
```