

## UNIDAD DIDÁCTICA N°15

Son autores de esta unidad didáctica:

Angel Sánchez Solanilla  
Máximo Bolea Campo  
Andrés Sánchez Otín

Coordinación pedagógica:

Carmen Candiotti López-Pujato

## I.- INTRODUCCIÓN



# I.- INTRODUCCIÓN

## PRESENTACIÓN

En esta unidad abordaremos el problema de controlar un ascensor de dos pisos. Analizaremos algunas estrategias metodológicas para introducir en el aula la programación en LOGO. Por último, una máquina que simula una embotelladora servirá de ocasión para profundizar el aprendizaje del uso de sensores digitales y el concepto de la realimentación

## 1.- OBJETIVOS

En esta unidad didáctica se pretende que el profesorado sea capaz de:

- Comprender y aplicar el concepto de realimentación.
- Utilizar pulsadores como sensores digitales y programa procedimientos para su control.
- Usar variables para almacenamiento temporal de datos.
- Dominar estrategias para la introducción de LOGO en el aula.

## 2.- CONTENIDOS

### II.- FUNDAMENTOS DE CONTROL.

Control de entradas y salidas digitales.

- a.- Descripción de la máquina
- b.- Finalidad de la máquina
- c.- Las conexiones
- d.- El algoritmo
- e.- Los procedimientos

### III.- MANOS A LA OBRA.

Manos a la obra presenta la propuesta del control de un ascensor de dos pisos.

#### 1.- EL ASCENSOR DE DOS PISOS

- a.- Planteamiento del problema.
- b.- Las conexiones
- c.- El programa.

#### 2.- REALIMENTACIÓN

- a.- Modificación de la construcción.
- b.- El nuevo algoritmo.
- c.- El nuevo programa.

#### IV.- CON NUESTROS ALUMNOS Y ALUMNAS

Orientaciones metodológicas para iniciar la enseñanza de LOGO en el aula.

- 1.- Preparación de las primeras actividades de programación.
- 2.- Presentación didáctica del modo directo.
  - 2.a. El aprendizaje de las primitivas.
  - 2.b. El cuaderno de LOGO
- 3. La utilidad didáctica de los mensajes de error.

#### V.- ENTRE MÁQUINAS Y HERRAMIENTAS

CONTROL CON SENSORES. PLANTA EMBOTELLADORA

- 1.- CONTROL DE UNA PLANTA EMBOTELLADORA
- 2.- CONEXIONES
- 3.- ORGANIGRAMAS, ALGORITMOS
- 4.- PROPUESTA DE MEJORA
- 5.- OTROS PROBLEMAS DE CONTROL

#### VI.- BIBLIOGRAFÍA

#### VII.- SOLUCIONES

#### 3.- CONOCIMIENTOS PREVIOS

Para poder introducir al alumnado en el uso del lenguaje Logo debe conocerse dicho lenguaje con bastante soltura y, sobre todo, su estructura. Para comprender y afianzar los conceptos y procedimientos vinculados con el control del ascensor y los nuevos contenidos del apartado "Entre máquinas y herramientas" debe conocerse el uso, conexión y funcionalidad de pulsadores, interruptores y conmutadores.

## II.- FUNDAMENTOS DE CONTROL

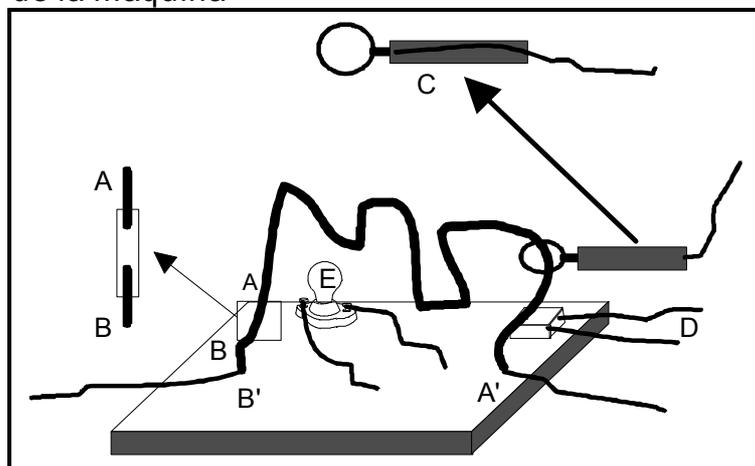


## II.- FUNDAMENTOS DE CONTROL

### CONTROL DE ENTRADAS Y SALIDAS

Pulso y habilidad podría llamarse el nuevo ejercicio de control que se presenta. Es un sencillo mecanismo que da pie para afianzar la programación del control de los actuadores y los sensores digitales. Por su fácil y rápida construcción puede realizarse en muy poco tiempo y utilizarse posteriormente como recurso para la motivación del alumnado.

#### a.-Descripción de la máquina



El dispositivo consiste en un circuito construido con alambre un poco grueso y algo retorcido (A' - B') sujeto por sus extremos a una plataforma. El alambre debe ser recorrido, sin ser tocado, con una anilla enmangada, desde la posición A' a la B'. La anilla es atravesada por el alambre.

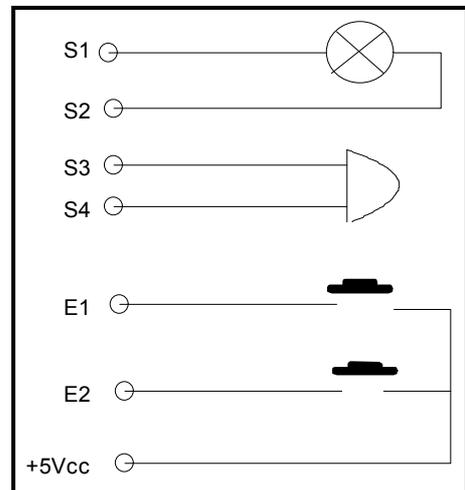
#### b.- Finalidad de la máquina

Se trata de un juego de habilidad; se debe recorrer el laberinto usando la anilla y sin tocar el alambre, pues de lo contrario se comete una falta; para indicar que se ha llegado al final sí debe tocarse con la anilla el alambre (zona B - B').

Los elementos del dispositivo son una bombilla (E), un zumbador o timbre (D), la anilla enmangada (C) y el alambre (A' - B'). Como puede apreciarse en la ampliación de una zona del alambre, se utilizan dos trozos pues es necesario determinar dos zonas; una, la más larga, que no se tiene que tocar (A' - A) y otra (B - B'), al final del recorrido, que sí debe tocarse para indicar que se ha llegado a dicho punto. Como ambos trozos no deben estar en contacto se han empalmado mediante un tubito de plástico.

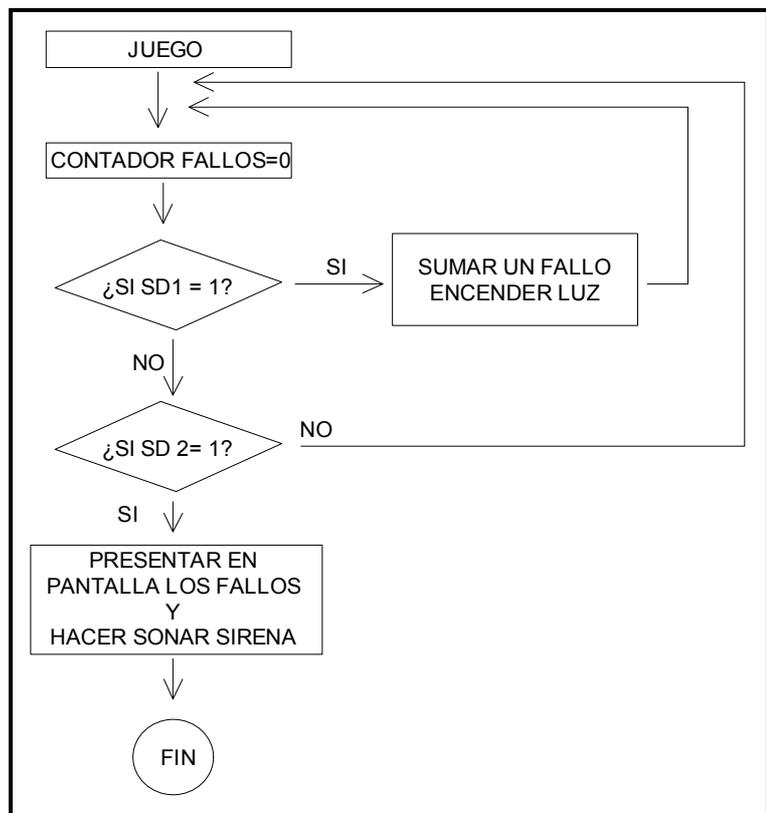
### c.- Las conexiones

Las conexiones son como las del gráfico de la derecha, pero deben explicarse detenidamente, sobre todo las de los sensores. La bombilla se conecta a S1-S2, el zumbador a S3-S4 y hay un sólo pulsador, la anilla. Si la anilla toca el alambre en la primera zona (A'-A), cierra E1 con +5vcc; y si toca en la zona (B-B'), cierra E2 con +5vcc. La anilla va directamente conectada a +5vcc, el segmento (A'-A) del recorrido del alambre a E1 y el segmento (B-B'), a E2.



### d.- El algoritmo

El comportamiento del dispositivo será el siguiente: si se toca con la anilla el alambre de la primera zona, se penaliza al jugador o jugadora con un punto y se enciende la luz un lapso de tiempo; en el monitor del ordenador aparecerá la puntuación alcanzada (cuantos menos puntos mejor); para indicar que se ha terminado la prueba, al llegar a la última zona, debe tocarse con la anilla el alambre (B-B'); se detendrá la ejecución del programa, sonará el zumbador y aparecerá en el monitor del ordenador la puntuación obtenida.



e.- Los procedimientos

PARA JUEGO

HAZ "CONTADOR 0;

Inicializa la variable llamada CONTADOR dándole valor 0

CONTROL;

Deriva a CONTROL

M2 "D ESPERA 200 M2 "P;

Activa el zumbador durante 2 segundos

ESCRIBE FRASE [FALLOS COMETIDOS:] :CONTADOR;

Se escribe en el área de textos la frase formada por la lista entre corchetes y el valor de la variable contador que son los fallos cometidos.

FIN

PARA CONTROL

SI SD 1= 1 [FALLO];

Si se activa el sensor 1 -la anilla toca el alambre de la primera zona- se deriva al procedimiento FALLO para contabilizarlo e iluminar la bombilla.

SI SD 2=1 [ALTO];

Si se activa el sensor 2 -la anilla toca la última zona- es señal de que se ha acabado el recorrido y se quiere terminar -se abandona el procedimiento actual y regresamos a JUEGO, que es el procedimiento del que se venía-.

CONTROL;

FIN

PARA FALLO

HAZ "CONTADOR :CONTADOR +1;

Hace la variable contador igual a su contenido más uno. Cuenta los fallos.

M1 "D ESPERA 100 M1 "P;

Conecta la bombilla durante 1 segundo.

FIN

ACTIVIDAD RECOMENDADA



Escribe estos procedimientos en el editor de LOGO, pulsa en el intérprete para que los compile el programa, sal del editor y, desde el área de trabajo, ejecútalos escribiendo JUEGO y pulsando [INTRO].



El árbol de procedimientos demuestra la relación entre los distintos procedimientos y permite compararlo con el algoritmo diseñado previamente para encontrar las desviaciones respecto a lo programado.



Se inicia el programa llamando al procedimiento JUEGO y este llama a CONTROL que, además de recursivo, puede saltar a FALLO si las condiciones especificadas se cumplen.



Envía un mensaje a la tutoría para aclarar tus dudas o, si este es el caso, para solicitar más información sobre los contenidos de este apartado.

### III.- MANOS A LA OBRA



### III.- MANOS A LA OBRA

#### CONTROL DE UN ASCENSOR DE DOS PISOS

En esta unidad didáctica, se propone la robotización del ascensor que ya se tenía construido; con ello, se pretende practicar el uso de la controladora y del micromundo de control. Se estudiarán, en un primer momento, las acciones de subida y bajada del ascensor para desarrollar, posteriormente, el concepto de la realimentación de una máquina o proceso robotizado usando más sensores digitales.

#### RECUERDA



- La realimentación es una de las características más importantes de cualquier proceso robotizado, que lo diferencia de cualquier otro relacionado con la automatización de máquinas y procesos.

#### 1.- EL ASCENSOR DE DOS PISOS

##### a.- Planteamiento del problema.

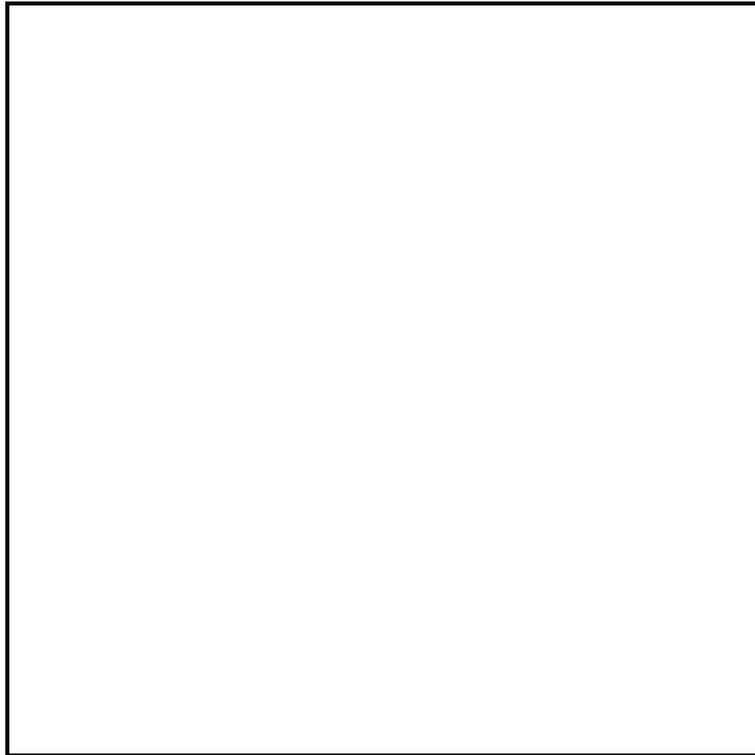
Partiendo del ascensor ya construido y cuyo esquema se presenta, la propuesta es la siguiente:

- 1.- Realizar un control simple de subida y bajada del ascensor de un piso a otro mediante dos pulsadores, uno para indicar la acción de ascenso y otro para producir el descenso.
- 2.- El paso posterior consistirá en incorporar la realimentación de las acciones de salida a través de entradas digitales, que permitan la lectura del estado de los pulsadores de llamada del ascensor y de llegada a cada piso.

##### Caso primero:

La cabina del ascensor se desplaza gracias a que está solidariamente fijada a la varilla roscada mediante unas tuercas. Al accionar el motor, éste transporta el movimiento a la polea fijada a la varilla que, al girar, hace que la cabina suba o baje en función del sentido de giro del motor.

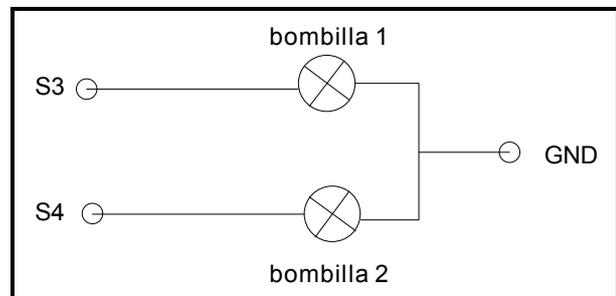
En un primer paso, lo que se pretende es que, al actuar sobre el pulsador de subida, el ascensor ascienda durante un período determinado de tiempo y se encienda una bombilla, y que al cerrar el pulsador de bajada la cabina descienda durante unos segundos, a la par que se conecta la bombilla indicadora de bajada.



Antes de continuar con la propuesta del ascensor, experimentemos una nueva forma de conectar dos bombillas a salidas digitales.

Realicemos las conexiones de las bombillas según el esquema siguiente:

Aprovechando el sistema utilizado por la controladora para activar las salidas digitales, conectamos dos bombillas a un par de estas salidas digitales, en este caso el par que se controla la primitiva M2 para que, al indicar una polaridad u otra, se encienda la bombilla 1 o la bombilla 2; ambas se conectan a masa.



PARA BOMBILLA1  
M2 "D  
ESPERA 300  
M2 "P  
FIN

PARA BOMBILLA2  
M2 "I  
ESPERA 300  
M2 "P  
FIN

PARA TOTAL  
M2 "D ESPERA 200  
M2 "P ESPERA 100  
M2 "I ESPERA 200  
M2 "P  
FIN

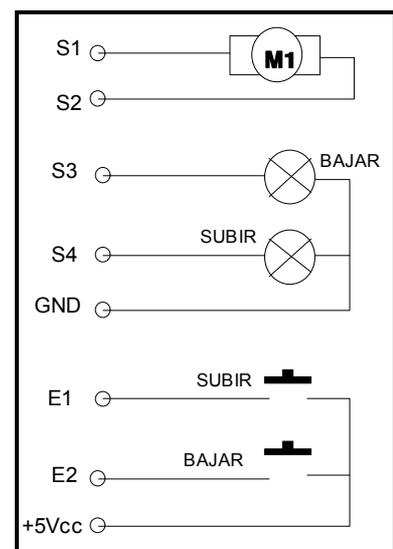
Es de suponer que, habiendo hecho la práctica anterior, se comprenderá fácilmente lo que se pretendía conseguir. Por lo tanto, para el control alternativo de dos bombillas conectadas a dos salidas digitales (correspondientes a un par controlable por la primitiva MOTOR) y a masa, puede procederse como si se tratara de un motor. Así como se activa una entrada u otra, según el sentido de giro que desee controlar, logramos encender una u otra bombilla, según cuál sea la salida activada.

#### b.- Las conexiones

Ahora se puede iniciar el control del ascensor. En primer lugar, se realizan las conexiones como se muestra en el dibujo.

Repitamos el proceso que se pretende llevar a cabo con el ascensor:

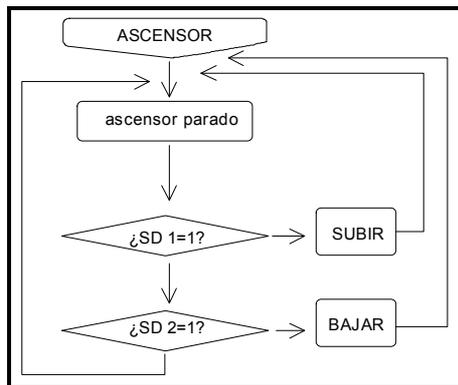
- 1º Al pulsar el sensor digital para subir, el motor hará ascender el ascensor durante un tiempo determinado y la bombilla señalada con la marca subir se mantendrá encendida hasta que se pare el motor.
- 2º La situación es semejante a la planteada en el punto primero pero, al pulsar el sensor para bajar, el motor gira en sentido



contrario durante un tiempo predeterminado y la bombilla reseñada como bajar se conecta durante el tiempo en que permanece activo dicho motor.

### c.- El programa

Descrito lo que se pretende, el proceso se puede plasmar en un algoritmo que responda a nuestra pretensión.



Lo primero es definir los procedimientos SUBIR y BAJAR. Debemos tener siempre presente la polaridad de conexión del motor; en nuestro caso, con la orden M1 "D el motor gira de tal forma que la cabina del ascensor sube; si este no es el caso, deben cambiarse los polos y, de esta forma los distintos procedimientos que se desarrollan a continuación servirán tal y como están .

Experimentemos con el procedimiento SUBIR para determinar el tiempo de espera más adecuado.

#### PARA SUBIR

M1 "D M2 "D ESPERA 100 M1 "P M2 "P;

M1 "D actúa sobre el motor y M2 "D sobre la bombilla que indica la acción de subir.

FIN

Ahora presentamos el procedimiento BAJAR.

#### PARA BAJAR

M1 "I M2 "I ESPERA 100 M1 "P M2 "P; M1 "I actúa sobre el motor y M2 "I sobre la bombilla que indica la acción de bajar.

FIN

Una vez determinados los tiempos de espera para ambos casos, integremos estos dos procedimientos en otro llamado ASCENSOR, que se ajuste al algoritmo anteriormente diseñado.

```

PARA ASCENSOR
SI SD 1 = 1 [SUBIR]
SI SD 2 = 1 [BAJAR]
ASCENSOR
FIN
    
```

Es el momento de probar el procedimiento ASCENSOR; si el ascensor se encuentra en el piso inferior, al pulsar el pulsador 1 el ascensor sube y después al pulsar el 2 desciende y así sucesivamente. Evidentemente, no deben pulsarse ambos sensores a la vez.

Este procedimiento deja sin resolver algunos problemas.

¿Cómo abandonar el programa?

¿Qué sucede si estando el ascensor en el piso superior volvemos a cerrar el pulsador 1 que inicia el procedimiento SUBIR?

¿Y si está en el inferior y se pulsa el sensor? Es mejor no probarlo, pues el motor giraría y podría averiarse el sistema.



- Modifica el procedimiento ASCENSOR para que se pueda interrumpir su ejecución sin usar la tecla ESCAPE. Envíalo a la tutoría.



Véase el vídeo donde se presenta la solución de la propuesta de trabajo que estamos analizando.

## 2.- REALIMENTACIÓN

Segundo caso.

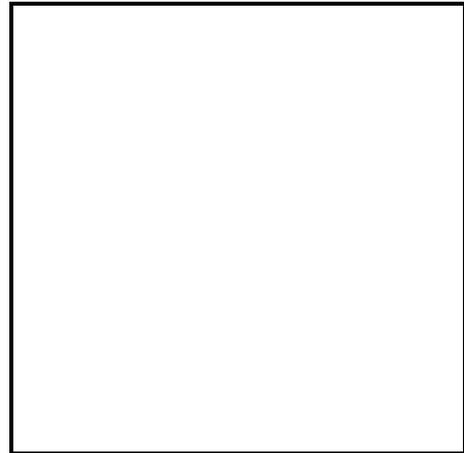
Para resolver el problema de subidas o bajadas indeseadas, es necesario dotar el sistema con los elementos necesarios para que el ordenador pueda detectar cuándo se pretende que suba el ascensor, aunque ya se encuentre en el piso superior, y no se active el motor. La misma solución habrá de arbitrarse para el supuesto de que el ascensor está en el inferior y se desee que baje.

Es necesario que el procedimiento LOGO pueda determinar en cada momento en qué piso se encuentra el ascensor. Habrá que dotarlo de los elementos mecánicos que

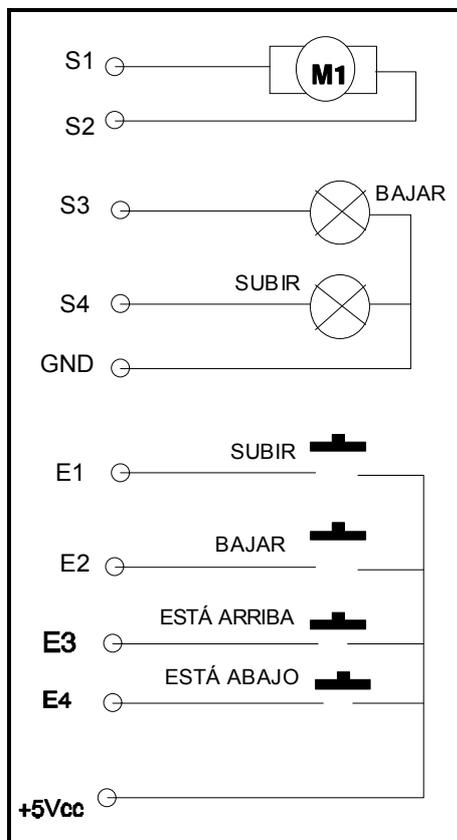
permitan al programa evaluar la posición de la cabina (planta baja o primer piso) y, por supuesto, hacer las modificaciones en el programa.

a.- Modificación de la construcción.

La modificación consiste en la colocación de dos sensores digitales (final de carrera, pulsador) en los extremos del recorrido de la cabina y un tope en la cabina que los oprima cuando se haya alcanzado la posición adecuada. Con estos dispositivos, analizando el estado de ambos sensores, se puede determinar la posición que ocupa el ascensor (planta baja o primer piso). Y de esta manera se controlan dos situaciones:



- a) Impedir que al llegar al piso deseado la cabina siga subiendo o bajando.
- b) Si la orden es ir al primer piso y ya se está en él, no se activa el motor y permanece parado.



El nuevo esquema de conexiones es el siguiente:

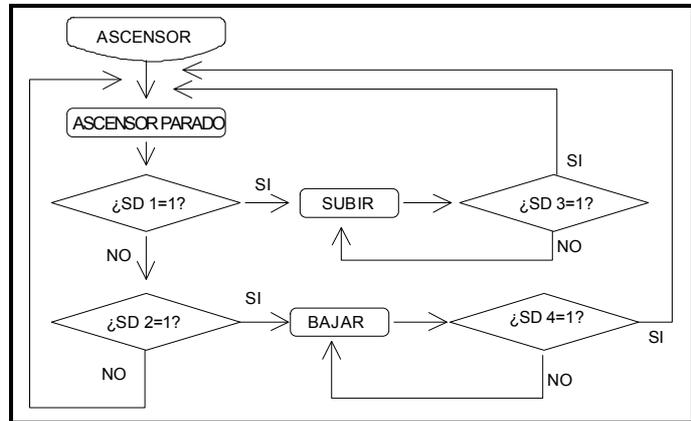
Los cambios descritos anteriormente quedan resueltos añadiendo dos sensores: el sensor (SD 3) para controlar si el ascensor está en el piso superior y el sensor (SD 4), utilizado para el control de la posición del ascensor en la planta baja. El resto de los elementos permanecen como estaban.

Ahora debemos pensar en el algoritmo que permita resolver el nuevo planteamiento.

Una solución al problema puede ser la que se presenta a continuación (se ha dicho en reiteradas ocasiones que las soluciones pueden ser muy variadas y que lo importante son los procesos puestos al servicio de su búsqueda).

b.- El nuevo algoritmo.

Se supone, inicialmente, que el ascensor está detenido en cualquier posición. Pueden producirse dos acciones: accionar el sensor 1 o el 2. Si se acciona el 1, se está ordenando al ascensor que suba, se inicia la subida y se comprueba si ya está en el piso superior (sensor 3 pulsado). Si lo está, detiene su acción y se retorna al inicio del ciclo; de no estarlo, sigue subiendo hasta que llega arriba y se cumple la condición de sensor digital pulsado (valor 1).



La segunda situación que puede darse es la del apretar el pulsador número 2 con la pretensión de hacer bajar al ascensor. Aquí se produce una situación similar a la de subida; se inicia el descenso analizándose el estado del sensor número 4; si está cerrado (valor 1) el descenso se detiene y, de lo contrario, se sigue descendiendo hasta que se cumple la condición de ser pulsado.

c.- El nuevo programa

PARA ASCENSOR

SI SD 1 = 1 [SUBIR];

Se comprueba el estado del pulsador 1 y, si está pulsado, se deriva el flujo al procedimiento SUBIR. Si no lo está, se pasa a ejecutar la siguiente línea del procedimiento actual.

SI SD 2 = 1 [BAJAR];

Se comprueba el estado del pulsador 2 y, si está pulsado, se deriva el flujo del programa al procedimiento BAJAR. En caso contrario, el flujo pasa a la siguiente línea.

ASCENSOR;

Recursividad. El procedimiento se llama a sí mismo. Vuelve a ejecutarse ASCENSOR.

FIN

#### PARA SUBIR

SI SD 3 = 1 [M1 "P M2 "P ALTO];

Se comprueba si el sensor digital 3 está pulsado (señal de que ya está arriba) y si lo está (valor 1), el motor se para, la bombilla se apaga y se abandona (ALTO) el procedimiento SUBIR y se regresa a ASCENSOR.

M1 "D M2 "D;

El motor y la bombilla se activan en un determinado sentido. M1 hace subir el ascensor y M2 conecta la bombilla.

SUBIR;

Recursividad. Se vuelve a ejecutar el procedimiento SUBIR

FIN

#### PARA BAJAR

SI SD 4 = 1 [M1 "P M2 "P ALTO];

Se comprueba si el sensor digital 4 está pulsado (señal de que ya está abajo el ascensor) y si lo está (valor 1), el motor se detiene, la bombilla se apaga y se abandona (ALTO) el procedimiento BAJAR. En caso contrario, se sigue con la siguiente línea del procedimiento.

M1 "I M2 "I;

M1 hace bajar la cabina del ascensor y M2 conecta la otra bombilla.

BAJAR;

Recursividad. Se vuelve a ejecutar el procedimiento BAJAR.

FIN

La realimentación, acción fundamental en el control, se pone de manifiesto en los procedimientos SUBIR y BAJAR. En ellos se está permanentemente chequeando el estado de los sensores y, en función del valor que tengan, se actúa de una forma o de otra. Este concepto y esta forma de diseñar los procedimientos será una constante en los distintos problemas por resolver en este curso.

#### ACTIVIDAD RECOMENDADA



Propuesta:

- Hacer las modificaciones necesarias en los procedimientos anteriores, con el fin de que se produzca una información en la pantalla que indique en qué piso se encuentra el ascensor



Envía a la tutoría los procedimientos con las modificaciones introducidas de acuerdo con lo indicado en la actividad recomendada precedente.



## IV.- CON NUESTROS ALUMNOS Y ALUMNAS



## IV.- CON NUESTROS ALUMNOS Y ALUMNAS

### ORIENTACIONES METODOLÓGICAS PARA INICIAR LA ENSEÑANZA DE LOGO EN EL AULA.

#### 1.- Preparación de las primeras actividades de programación

En ocasiones anteriores de este curso, hemos hecho un símil entre las funciones de relación de los seres vivos y el funcionamiento del ordenador para facilitar con dicha estrategia a alumnos y alumnas la comprensión del papel del ordenador en el control.

En la unidad didáctica XIV, hemos descrito los elementos que intervienen en los procesos de robotización: La unidad central como elemento de control principal, los periféricos que le transmiten información o la reciben, y ejecutan las órdenes que la unidad central les imparte.

Vamos a centrarnos, en primer lugar, en la presentación didáctica de la unidad central; convendrá subrayar, al desarrollar el tema con los alumnos, que ésta es el lugar donde todos los periféricos están conectados, puestos a su servicio para que los controle por medio de un programa que se diseñará gracias a un lenguaje de programación.

Nuestro trabajo va a consistir en orientar los aprendizajes para que el alumnado aprenda a diseñar el "pensamiento" de la máquina, es decir, crear una relación causa-efecto entre los datos que la máquina objeto de control recibe por medio de los sensores, y la respuesta que va a dar poniendo en marcha los actuadores (motores, bombillas, electroimanes, etc.)

En la unidad didáctica XIV ya expusimos que para poder diseñar un procedimiento de control, un conjunto de decisiones, debe establecerse una comunicación con la unidad central; para lograrlo, contamos con un lenguaje común, LOGO. Como todo lenguaje, LOGO es un código arbitrario formado por un vocabulario, sometido a una ortografía y a una sintaxis. Esos conceptos resultarán familiares a la clase porque deben haber sido desarrollados en el área de Lengua, Idioma Extranjero y Matemáticas.

Alumnos y alumnas deberán estar en condiciones de asimilar el siguiente proceso: mediante el lenguaje LOGO se crea un programa, un conjunto ordenado de instrucciones por medio del cual la máquina captará los datos aportados por los sensores, operará con ellos dándoles un significado asignándoles un valor, los comparará con otros preestablecidos para ordenar, por medio de una sentencia condicional, la puesta en marcha de unos actuadores u otros.

La parte fundamental de un programa residirá precisamente en las sentencias condicionales, es decir, en la capacidad de elección. A ellas dedicaremos otras unidades didácticas, pues ahora debemos solventar un problema didáctico previo: dar a conocer al alumnado el lenguaje LOGO, su vocabulario y su sintaxis. Comenzaremos nuestra exposición del modo más interactivo e inmediato: el modo directo.

## 2.- Presentación didáctica del modo directo

Antes de usar LOGO para el control de una máquina, nuestro alumnado deberá conocer los rudimentos de este lenguaje de programación. La forma más sencilla, directa y atractiva de proponerles su conocimiento será a través del micromundo de la tortuga, donde las órdenes producen efectos geométricos inmediatos. Aprovecharemos el primer contacto del alumnado con LOGO para hacerles vivenciar esta característica. Para conseguirlo, una vez cargado el programa y visualizada la pantalla con sus tres ventanas, les propondremos la siguiente actividad para la primera sesión:

Comunicarse con la tortuga por medio del teclado para que realice dibujos en la pantalla.

Les informaremos de que:

- La tortuga entiende órdenes en castellano.
- Las órdenes se expresan con una palabra.
- A veces, la orden va acompañada de un parámetro numérico que indica cantidad.
- Cuando no comprende un mensaje, LOGO lo indica por medio de otro mensaje en la ventana de textos.
- En cuanto descubran alguna palabra del vocabulario de la tortuga, la comunicarán a toda la clase escribiéndola en la pizarra.

Durante el desarrollo de la sesión, la función del profesor o profesora será ante todo de animación, ayudando al grupo a reflexionar sobre los distintos mensajes de error que irán recibiendo, de modo que distingan entre:

- NO SE CÓMO HACER CORRE - y - FALTAN DATOS PARA AVANZA -

De este modo, alumnos y alumnas irán haciendo deducciones que les ayudarán a descubrir algunas primitivas del lenguaje LOGO. Cuidaremos de que vayan descubriendo la función de cada ventana; así, en la ventana de GRÁFICOS verán que la tortuga dibuja pero, en la de TEXTO, LOGO envía mensajes mientras, los nuestros aparecen en la de TRABAJO.

Les haremos reflexionar sobre el carácter estricto del lenguaje LOGO en el que hasta los espacios en blanco tienen un valor y donde no caben interpretaciones que completen una palabra mal escrita según su contexto de uso, como ocurre con los lenguajes naturales.

Los giros de la tortuga son difíciles de descubrir, en algunos casos, porque engloban en un solo término la orden y el sentido de giro de la tortuga una vez ejecutada la orden. Ayudaremos a descubrir uno de ellos escenificando una orden de giro con un alumno y viendo la necesidad de darle un sentido además de un parámetro.

El juego no tiene por qué durar más de 30 minutos, pues una vez asumida la arbitrariedad del lenguaje, el juego se transforma en una lotería en la que el profesor o la profesora es el único que conoce los resultados.

#### b.- El aprendizaje de las primeras primitivas

El planteamiento más heurístico del aprendizaje de LOGO tendría que partir del descubrimiento individual o, mejor, grupal de su vocabulario y sintaxis. Sin llegar a extremos, podemos plantearnos presentar las primitivas al grupo conforme las vayan necesitando para solventar los problemas que se les plantean al diseñar los dibujos. De este modo, el papel del profesorado, además de animar y motivar, supondrá proporcionar información y será, así, un apoyo al que se recurre ante las dificultades.

También conviene pensar cuándo conviene que sean los propios alumnos y alumnas quienes piensen los diseños que quieren efectuar ( proyectos abiertos) o sea el profesor quien dé las ideas (proyectos libres) o presente un diseño concreto para que los equipos intenten lograr otro igual ( proyectos cerrados).



En el la unidad didáctica XII se ha propuesto un orden de presentación para las primitivas de LOGO. Seguiremos dicho orden aunque cualquier otro podría ser igualmente válido, siempre que partiera de una disposición activa y participativa por parte del alumnado.

Después de dedicar un rato a la investigación del vocabulario de LOGO, nos aseguraremos de que todo el alumnado conoce la sintaxis de estas primitivas:

AVANZA N  
RETROCEDE N  
GIRADERECHA N  
GIRAIZQUIERDA N

Sin necesidad de otro vocabulario propondremos que averigüen las dimensiones del terreno que visualizan en la pantalla de Gráficos, medidos en "pasos de la tortuga".

Valdrán actividades como:

- Dibujar una cruz en el centro de la pantalla lo más grande posible pero sin salirse por los laterales.
- Hacerle un marco a la ventana de gráficos.
- Dibujar un rectángulo que ocupe un cuadrante de la ventana
- Dibujar los cuatro cuadrantes de la ventana.
- Dibujar un camino, lo más recto posible, que vaya de extremo a extremo de la ventana.

La realización de estos proyectos llevará a la necesidad de tener limpia la pantalla para empezar un nuevo trabajo. Probablemente, nuestro alumnado demandará que le informemos de la primitiva que realiza esta tarea. Presentaremos, entonces, la primitiva BORRAPANTALLA.

Habrá llegado el momento de comunicarles que muchas de las primitivas que entiende la tortuga las puede comprender también en forma abreviada; AV, RE, GD, GI, BP.

Uno de los problemas que suelen surgir en estas primeras sesiones es el de dificultad de los alumnos para comprender que las órdenes GD y GI están en función de la posición de la tortuga y no de la del programador.

Aparte de hacerles notar que nuestra visión de la tortuga es cenital seguiremos la siguiente escenificación:

- Organizar un juego en el que un miembro del grupo actúe de tortuga ( puede llevar una flecha que indique la dirección y el sentido de su posición).
- El resto de su equipo diseña en un folio de papel cuadriculado un sencillo dibujo que marque una trayectoria a través de las mesas y sillas de la clase de modo que, al final, tenga que volver al lugar y a la posición de partida.
- El equipo irá dando órdenes LOGO en las que los cuadros del papel coincidan con las baldosas de la clase.

Valdrá la pena insistir en el concepto antes comentado (GD y GI en función de la posición de la tortuga), así como en los "mensajes de error" que toda la clase coreará cuando olviden los "programadores" de dictar un espacio o no indiquen un argumento.

Sabemos que LOGO OFRECE la posibilidad de repetir una orden o modificarla con sólo subir el cursor a la línea ya ejecutada y volver a pulsar [INTRO]. Podemos presentar a nuestro alumnado esta propiedad de LOGO escribiendo en la pizarra el nombre de una orden incorrectamente, como SUBE LAPIZ.

Una vez escrita y ejecutada, nos aseguraremos de que lean el mensaje de error. Subirán el cursor y modificarán la orden suprimiendo el espacio intermedio y comprobando, después de pulsar [INTRO], que aparentemente tampoco pasa nada, (no aparece tampoco mensaje de error). Será, pues, al ordenar el movimiento de la tortuga, cuando apreciarán por medio de esta primitiva que la tortuga puede desplazarse sin dejar rastro. La primitiva opuesta podrán deducirla por sí mismos, así como GOMA, instrumento de uso cotidiano.

Dentro de las posibilidades de edición inmediata que nos ofrece LOGO, advertiremos que se pueden escribir varias órdenes separadas por espacios dentro de una misma línea, que no tiene por qué acabar en el extremo de la ventana. Propondremos como ejemplo la sucesión AV 50 GD 90 de modo que, al repetirla cuatro veces, consigan un cuadrado.

Usando el modo directo y subiendo el cursor para repetir o modificar órdenes, propondremos los siguientes ejercicios:

- Dibujar las cifras arábicas en la pantalla.
- Situar la tortuga en un ángulo de la pantalla de gráficos para dirigirla posteriormente al otro extremo, trazando en el recorrido una línea quebrada, de forma que cada segmento concatenado sea cada vez menor y los ángulos cada vez mayores.
- Trazar una línea quebrada espiral desde el centro de la ventana, de modo que no se corte a sí misma.
- Realizar una espiral doble que, una vez se ha alejado del centro intente regresar por el pasillo que ha dejado el trazo anterior.
- Trazar dos segmentos paralelos.

Los ejercicios darán pie para presentar las primitivas CENTRO y LIMPIA por la conveniencia de tener controlada la posición de la tortuga antes de programar una serie de movimientos, pues suele suponer un serio tropiezo en el aprendizaje de la programación por parte de nuestros alumnos y alumnas. Un simple giro mal concebido cambia totalmente el aspecto de su diseño y provoca una gran frustración.

## 2.b. El cuaderno de LOGO

Dentro de este apartado expondremos una serie de tareas cuyo desarrollo supone el ejercicio del control de la tortuga, así como el paso previo a la creación de procedimientos. Para la realización de estos ejercicios, pediremos a nuestro alumnado que dibuje en un cuaderno de papel cuadriculado el diseño que desea realizar, así como el conjunto de órdenes necesario para ello. Después, se trabaja con LOGO.

No se valorarán los sucesivos errores cometidos hasta llegar al diseño final, ya sea corrigiendo con la primitiva GOMA, ya sea retrocediendo, cambiando el ángulo de giro o haciendo tanteos a pasos cortos. Al final, quedará en la pantalla de trabajo una serie de órdenes separadas por espacios en blanco, que comenzarán con un BP y terminarán con el cursor al final de la línea. Cuando hayan llegado a esa situación, tanto en la pantalla como en su cuaderno, llamarán al profesor/a para que sea testigo de la ejecución de las órdenes, con la esperanza de que el diseño del cuaderno se repita en la pantalla.

El cuaderno sirve también como diccionario LOGO y puede ser usado por el profesorado como recurso de control del trabajo diario realizado en el aula por los distintos equipos.

Otros ejercicios útiles para afianzar el aprendizaje de las primitivas son los siguientes:

- Dibujar una espiral quebrada cuadrada con una sucesión de avances que aumenten la longitud de los lados.
- Hacer que la tortuga dibuje la palabra LOGO.
- Dibujar unas gafas de cristal rectangular en la pantalla.

A lo largo de la sesión, insistiremos a nuestro alumnado en la necesidad de adquirir algunos hábitos muy útiles en la programación:

- La necesidad de simplificar los órdenes para visualizar rápidamente los errores; así, dos avances seguidos se transforman en su suma; dos giros opuestos se anulan; antes retroceder que girar 180°.
- La necesidad de tener controlada la tortuga para evitar que se salga de la ventana o dibuje formas no deseadas.
- El interés por la búsqueda de la figura precisa, limpia y centrada en la ventana sin conformarnos con el "ya vale".

A estas alturas de conocimiento de LOGO, sería conveniente que ya hubiera surgido la necesidad de hacer uso de primitivas como :

PONCOLORLAPIZ N,  
PONFONDO N,  
PONGROSOR N,  
RELLENA

### 3.- La utilidad de los mensajes de error

No se debería terminar sesiones como la descrita sin hacer una mención expresa a la responsabilidad última del programador en la bondad o limitaciones del programa diseñado, introducido y ejecutado.

¿Quién es el responsable de que una máquina diseñada para pintar paredes, manche el suelo en lugar de extender la pintura por los tabiques?, ¿La máquina o la persona que la diseñó? En el supuesto de un programa creado por nosotros, debemos tener presentes las limitaciones de los datos ofrecidos por los sensores, la arquitectura de la máquina diseñada... Por último, si no actúa tal y como queríamos que actuara, el error es nuestro y debemos corregirlo.

El profesor o la profesora hará notar a su alumnado que:



- El error no es un castigo sino una comunicación del programa al programador que indica qué es lo que debe evitar en el futuro.
- La energía generada por la frustración ante el error debe llevarles hacia la superación por medio de la comunicación dentro de los grupos de trabajo.

Sería positivo que la dinámica del trabajo de cada grupo permitiera recoger, en el cuaderno de trabajo de los distintos equipos, las sucesivas creaciones, ensayos y errores para aprender de sus propios fallos. El profesor o la profesora, en consecuencia, debiera aparecer como un reconductor que ayuda a relativizar los fallos y encauza la actividad y el debate dentro del grupo para que sea éste quien alcance sus objetivos.



Deja un mensaje en la tutoría comentando tus impresiones acerca de las estrategias desarrolladas es este apartado.

## V.- ENTRE MÁQUINAS Y HERRAMIENTAS



## V.- ENTRE MÁQUINAS Y HERRAMIENTAS.

### CONTROL CON SENSORES. PLANTA EMBOTELLADORA.

#### 1.- Control de una planta embotelladora.

Nos proponemos continuar con el aprendizaje del control de máquinas movidas por motores. En el presente ejemplo, introduciremos los sensores (pulsadores) con objeto de obtener un control retroalimentado de la máquina. Los programas de control que desarrollaremos activarán los motores en función del estado de estos sensores.

Partimos de la planta embotelladora que nos proporcionan los "kits" de montaje Fischer.



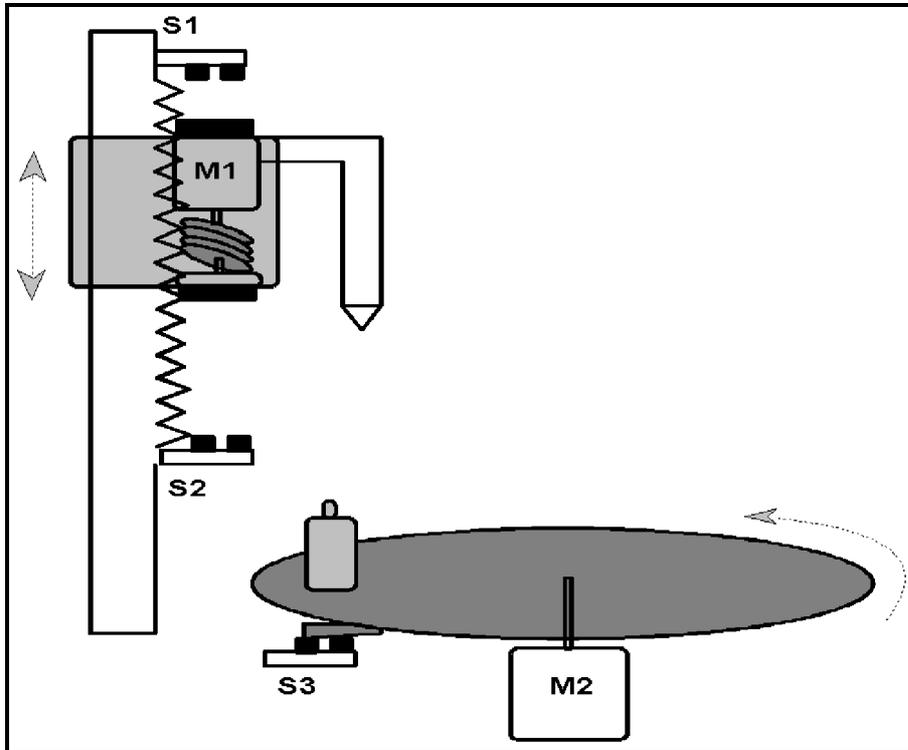
Véase el vídeo correspondiente a esta unidad, sección "Entre máquinas y herramientas"

La planta embotelladora dispone de dos motores, controlados por las primitivas:

- M1, para el ascenso y descenso del brazo que sella las botellas.
- M2, que hace rotar la plataforma que contiene las botellas.

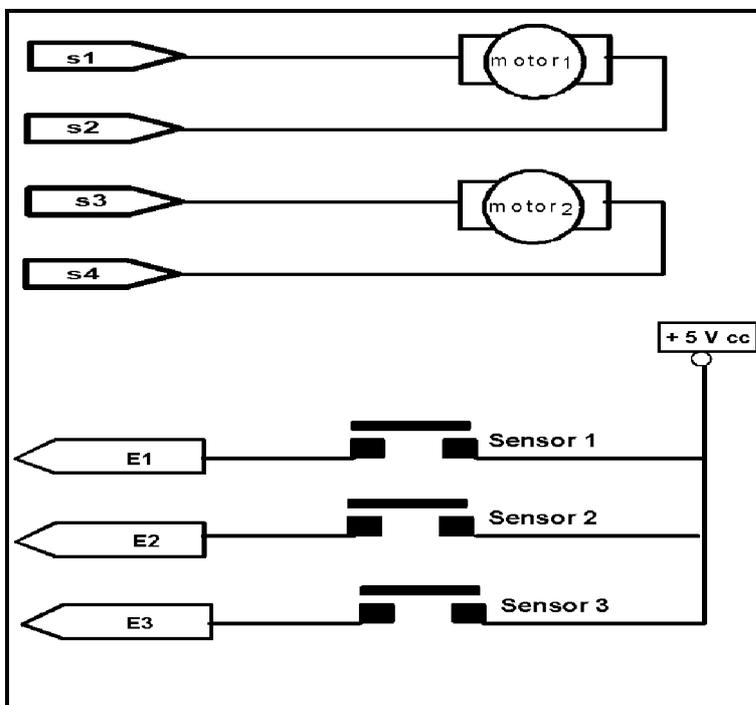
Además, disponemos de tres sensores digitales (pulsadores):

- S1, Situado en la parte superior del brazo vertical de la embotelladora, determinará el límite superior del recorrido del cabezal embotellador que mueve el motor M1.
- S2, Situado en la parte inferior del brazo vertical de la embotelladora, determinará el límite inferior del recorrido más bajo del cabezal embotellador que mueve el motor M1.
- S3, Tangente a la plataforma giratoria que porta las "botellas", mediante las pletinas metálicas instaladas en el borde de esta plataforma, determinará la posición exacta de su parada con objeto de permitir al brazo embotellador actuar directamente sobre la "botella".



Esquemáticamente se muestra la estructura de la embotelladora y la ubicación de sensores y actuadores. Paso a paso, trataremos de llevar a término el control regulado de esta planta embotelladora.

## 2.- Conexiones



Conectaremos el motor 1, que sube y baja el brazo embotellador, a las salidas digitales S3-S4 de la controladora.

Conectaremos el motor 2, que hace girar la plataforma, a las salidas digitales S1-S2 de nuestra controladora.

Conectaremos el sensor 1, fin de recorrido superior del brazo, a +5Vcc y la entrada digital E1 de la controladora.

Conectaremos el sensor 2, fin de recorrido inferior del brazo, a +5Vcc y la entrada digital E2 de la controladora.

Por último, conectaremos el sensor 3, que delimita las posiciones de parada del giro de la plataforma, a +5Vcc y la entrada digital E3 de la controladora. Todo ello como indica el esquema anexo.

Realizado el montaje, encenderemos la controladora y el ordenador y cargaremos Windows, Win-Logo y el micromundo de control de la controladora "CONTROLA, como de costumbre.

Pasaremos, a continuación, a probar en modo directo el montaje y, si es preciso, realizaremos los ajustes pertinentes en la posición de nuestros sensores.

Convengamos que el giro a la derecha del motor\_1 implica el ascenso de la embotelladora, ejecutaremos desde la ventana de trabajo:

M1 "D ESPERA 200 M1 "P ; La embotelladora subirá durante 2 segundos. Convengamos, por consiguiente, que el giro a la izquierda del motor\_1, implica el descenso de la embotelladora; ejecutaremos desde la ventana de trabajo:

M1 "I ESPERA 200 M1 "P ; La embotelladora bajará durante 2 segundos.

Comprobemos el giro de la plataforma con el segundo motor:

M2 "D ESPERA 300 M2 "P ; La plataforma girará durante 3 segundos.

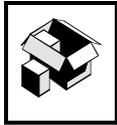


Si has tenido alguna dificultad, después de haber procurado solucionarla, ponte en comunicación con la tutoría para solicitar orientación e información para seguir avanzando.

### 3.- Organigramas, algoritmos.

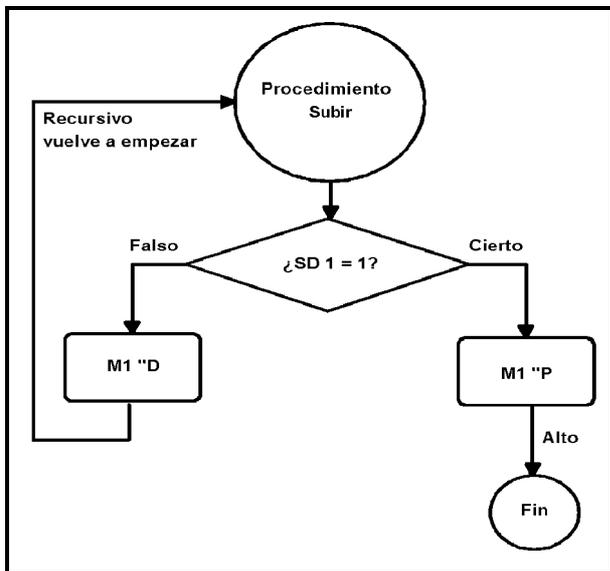
El carácter modular de LOGO hace posible abordar la programación que dé solución a aspectos parciales del problema de control planteado. En primer lugar, crearemos un procedimiento para que el brazo embotellador ascienda.

Podremos saber que el brazo ha alcanzado su recorrido superior en la ascensión a través del sensor digital 1. Si el brazo ha alcanzado esta posición, el pulsador estará cerrado y la lectura del sensor digital con la primitiva SD 1 nos devolverá 1; en cualquier otro momento del recorrido, el pulsador estará abierto y la lectura del sensor nos devolverá 0.



Repasa el tratamiento recursivo y el uso de condicionales para finalizar los procedimientos recursivos. Están explicados en esta misma sección, en las unidades didácticas anteriores.

El procedimiento para el control del brazo puede ser como sigue:



Si el sensor digital 1 no señala el contacto de fin del recorrido superior, devolverá 0 y hará falsa la condicional. En consecuencia, se seguirá activando el motor 1 (en modo directo, ascensión) y se volverá a iniciar recursivamente el procedimiento. Por el contrario, si el sensor digital 1 señala el contacto de fin del recorrido superior, devolverá 1 y hará cierta la sentencia condicional. En consecuencia, se parará el motor 1 y finalizará el procedimiento.

#### PARA SUBE

SI SD 1 = 1 [M1 "P ALTO] ;

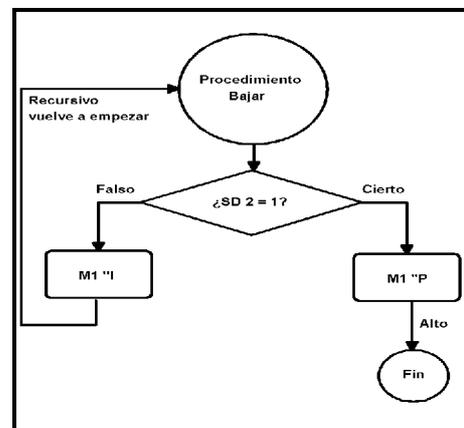
Si el sensor 1 devuelve 1, se para el motor y es el fin del procedimiento.

M1 "D SUBE ;

Activa el motor 1 (subiendo) y el procedimiento vuelve a empezar: llamada recursiva.

FIN

Conviene implementar el procedimiento SUBE y ejecutarlo; se observará que el brazo de la embotelladora sube hasta el fin de su recorrido y se detiene. Análogamente, el diagrama para el control del sensor 2 puede ser como sigue:



Si el sensor digital 2 no señala el contacto de fin del recorrido inferior, devolverá 0 y hará falsa la sentencia condicional. En consecuencia, se seguirá activando el motor 1 (en modo inverso, descenso) y se volverá a iniciar recursivamente el procedimiento. Por el contrario, si el sensor digital 2 señala el contacto de fin del recorrido superior, devolverá 1 y hará cierta la sentencia condicional. En consecuencia, se parará el motor 1 y finalizará el procedimiento.

PARA BAJA

SI SD 2 = 1 [M1 "P ALTO] ;

Si el sensor 2 devuelve 1, se para el motor y finaliza el procedimiento.

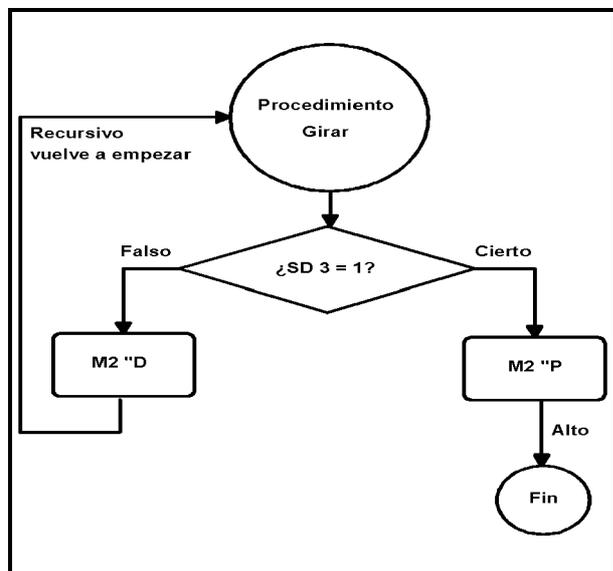
M1 "I BAJA ;

Activa el motor 1 (bajando), y el procedimiento vuelve a empezar: llamada recursiva.

FIN

Es aconsejable implementar el procedimiento BAJA y ejecutarlo; se observará que el brazo de la embotelladora baja hasta el fin de su recorrido y se detiene.

Del mismo modo se procede para controlar el sensor 3:



Si el sensor digital 3 no señala el contacto con alguna pletina de la plataforma giratoria, devolverá 0 y hará falsa la condicional. En consecuencia, se seguirá activando el motor 2 y se volverá, recursivamente, a iniciar el procedimiento. Por el contrario, si el sensor digital 2 señala un contacto (cerrado), devolverá 1 y hará cierta la sentencia condicional. En consecuencia, se parará el motor 2 y finalizará el giro de la plataforma y el procedimiento.

PARA GIRA

SI SD 3 = 1 [M2 "P ALTO] ;

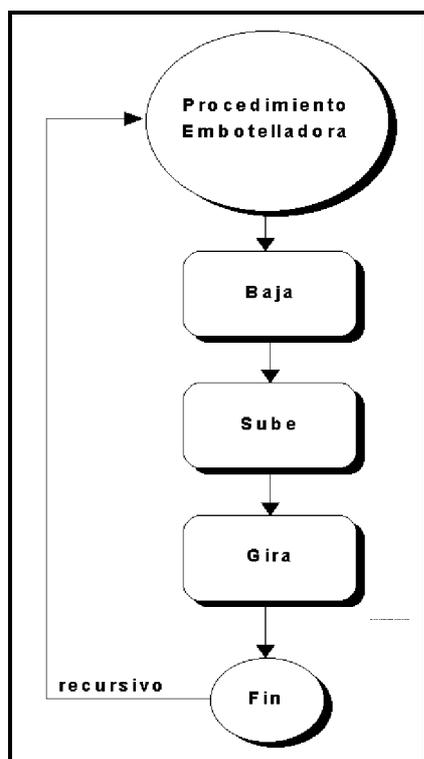
Si el sensor 3 devuelve 1, se para el motor 2 y finaliza el procedimiento.

M2 "D SUBE ;

Activa el motor 2 (gira la plataforma), vuelve a empezar: llamada recursiva

FIN

Procúrese implementar el procedimiento GIRA y ejecutarlo en modo directo; se observará que la plataforma gira hasta colocar la "botella" debajo del brazo y luego se detendrá. Si es preciso, ajústese la posición del sensor 3.



Para hacer funcionar toda la planta necesitamos reunir estos procedimientos en uno, de tipo recursivo, que los ejecute secuencialmente.

El brazo deberá bajar para colocar el "tapón" a una "botella", luego deberá subir para permitir el giro a la plataforma, y una vez que ésta gire, una nueva "botella" estará en la posición apropiada. El procedimiento deberá volver a iniciarse.

#### PARA EMBOTELLADORA

BAJA ; El brazo baja.  
 SUBE ; El brazo sube.  
 GIRA ; La plataforma gira.  
 EMBOTELLADORA ; Llamada recursiva, vuelve a empezar.

FIN

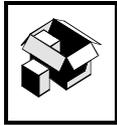
Ha llegado el momento de introducir este procedimiento en la ventana de trabajo de Win-Logo y ejecutarlo. ¡Ya está en funcionamiento la embotelladora!



Ponte en comunicación con al tutoría e informa de los resultados obtenidos en esta práctica guiada.



Recuerda que para salir de un procedimiento recursivo que no tiene una condición de salida se debe de pulsar la tecla [ESC]



Repasa la instrucción condicional, ya tratada en esta sección en unidades didácticas anteriores: SI condición [acciones\_caso\_cierta] [acciones\_caso\_falsa] , así como la función primitiva TECLA?

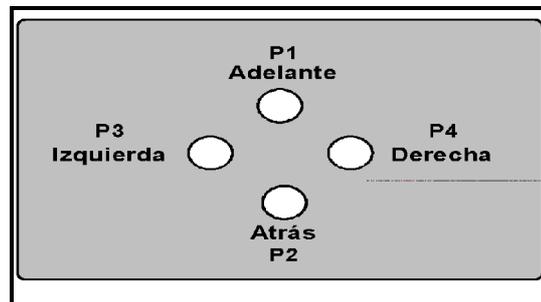
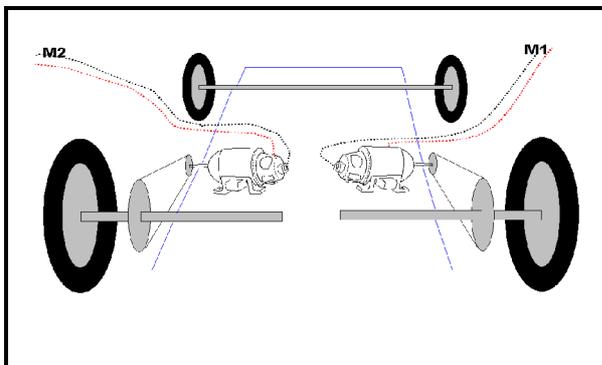
Otros problemas de control



Control de un móvil



En la unidad didáctica anterior, en esta misma sección (Apdo. 3), desarrollamos el control de un móvil impulsado por dos motores, acoplados uno a cada rueda del tren delantero. Además, establecimos su control mediante un conjunto de cuatro sensores digitales para realizar el avance, retroceso y los giros a izquierda y derecha.



## ACTIVIDAD RECOMENDADA

Proponemos ahora su ampliación con un sensor digital en el parachoques delantero de modo que, tras el choque con un obstáculo, sea capaz de retroceder y girar 180 grados.



Envía a la tutoría el diagrama y los procedimientos de control que dan solución al primer problema planteado.

## ACTIVIDAD 1

Disponemos de un aparcamiento con 100 plazas de aparcamiento, deseamos instalar un sistema de control que nos proporcione la información de si está completo o por el contrario quedan plazas libres.

Desarrolla un sistema de control para que nos proporcione la información de si el aparcamiento está "completo" o "libre" mediante las luces roja o verde de un semáforo.

## VII.- SOLUCIONES



## VII.- SOLUCIONES

### ACTIVIDAD 1

El problema consiste en desarrollar un sistema de control que informe si un aparcamiento está completo o hay plazas libres mediante un semáforo.

#### 1.- Entradas

Una primera idea consiste en utilizar sensores en el suelo de forma que el peso de los automóviles en las plazas ocupadas los cierre, en tanto que los de las plazas libres quedaran abiertos.

Podríamos sentirnos tentados de conectar cada sensor a una de las entradas digitales de la controladora; así, mediante la lógica de la programación podríamos disponer de la información requerida. Obviamente, no podemos hacerlo pues no disponemos de tantas entradas en la controladora.

Los propios sensores se comportan conforme a la lógica; puestos en serie operan como el operador AND (conjunción) y en paralelo como el OR (disyunción).

Así, sólo si las 100 plazas están ocupadas, se cerrarán todos los pulsadores permitiendo el paso de corriente. En este caso, los 100 pulsadores  $P_i$  estarán en estado 1 y:  $1 \text{ AND } 1 \text{ AND } 1 \text{ AND } 1 \text{ AND } \dots\dots\dots 1 \text{ AND } 1 = 1$

responderán como un único sensor digital, el S1, que nos proporcionará la información requerida.

#### 2.- Las salidas:

Una forma de reflejar la información de salida mediante el control de dos bombillas, verde una y roja la otra, se indica en la figura adjunta. Proponemos su simplificación. Las primitivas del micromundo de control M1 para aparcamiento libre y M2 para completo podrían servirnos para proporcionar esta información.

El procedimiento de control resulta sencillo.

PARA GARAJE  
SI  $SD = 1$  [M1 "P M2 "D] [M1 "D M2 "P]  
GARAJE  
FIN

La primitiva SI (condicional) ya es conocida. Si todas las plazas están ocupadas, todos los pulsadores estarán cerrados y el sensor digital devolverá 1; la condición será cierta y se ejecutará la primera lista: se apagará la bombilla verde (Motor\_1 Parado) y se encenderá la roja (Motor\_2 Derecha), indicando así que está completo. Si falta algún vehículo, algún pulsador quedará abierto y el sensor digital S1 devolverá 0. En consecuencia, se ejecutará la segunda lista encendiendo la luz verde y apagando la roja para indicar así que está libre.

## MEJORAS

Una alternativa para evitar la infraestructura de instalación de tantos sensores, suponiendo que la entrada al garaje y salida del mismo se efectúan por distintas puertas, consiste en poner un pulsador en la entrada, que se controla con la primitiva SD 1, y otro pulsador en la salida, que se controla con la primitiva SD 2. Así, podríamos llevar la cuenta de los coches que entran y salen y, por tanto, conocer si quedan plazas libres. El pulsador, que respondería al peso de un vehículo y no al de una persona, podría sustituirse por una célula fotoeléctrica que leyera la interrupción de un haz luminoso al paso de los vehículos.

El programa de control de esta situación no resulta complicado. Debemos definir una variable acumuladora para llevar la cuenta de los vehículos que hay en el garaje. Cuando el sensor digital de entrada marque una entrada, le sumaremos una unidad y si lo hace el de salida, se la restaremos.

Debemos tener cuidado de que el programa anote o descuenta las entradas o salidas al liberarse el pulsador. Podría suceder que un vehículo estuviera más tiempo del calibrado y lo contáramos varias veces.

Para evitar el problema del tiempo indefinido de actuación sobre el sensor, nos interesarán más las variaciones de los estados de los sensores digitales que los propios estados. Cada vez que entre un coche, SD 1 pasará del estado 0 al 1 al pisar el pulsador, y del estado 1 al 0 al liberarlo. Por cada coche que entre, SD 1 habrá experimentado dos variaciones de estado.

Análogamente cada vez que salga un coche, el SD\_2 pasará del estado 0 al 1 al pisar el pulsador y del estado 1 al 0 al liberarlo. Por cada coche que salga, el SD\_2 habrá experimentado dos variaciones de estado.

Podemos crear una variable acumuladora, que llamaremos ACUMULA para sumar en ella todas las variaciones del sensor de la entrada y restar las variaciones del sensor de salida. Si el número de coches es igual a 100, cuando ACUMULA valga 200 el aparcamiento estará lleno.

Para comparar las variaciones de estado de cada sensor necesitaremos sendas variables donde anotar los estados anteriores y los recién leídos. Denominaremos PESD1 y PESD2 los pasados estados de los sensores 1 y 2 y NESD1 y NESD2 los nuevos valores de esos sensores recién leídos.



Repasa la asignación de valores a variables. Puede verse en la unidad didáctica anterior, en el apartado II, Fundamentos de LOGO, de la unidad didáctica XII.

Para INICIAR el parquímetro necesitaremos asignar valor 0 a ACUMULA y dar valores nulos a los estados pasados de los sensores PESD1 y PESD2. A partir de ese momento, podremos iniciar el ciclo recursivo de control CICLOCONTROL.

```
HAZ "ACUMULA 0 HAZ "PESD1 0 HAZ "PESD2 0 CICLOCONTROL
```

Este ciclo recursivo, CICLOCONTROL, deberá:

- i) Leer el estado de ambos sensores y anotarlos como nuevos valores en sus variables NESD1 y NESD2:

```
HAZ "NESD1 SD 1 HAZ "NESD2 SD 2
```

- ii) Compararlos con los pasados estados y si hay variaciones anotarlas en ACUMULA.

Variación del sensor de entrada SD\_1 (positiva):  
SI NO (:NESD1 = :PESD1) [HAZ "ACUMULA :ACUMULA+1]

Variación del sensor de salida SD\_2 (negativa):  
SI NO (:NESD2 = :PESD2) [HAZ "ACUMULA :ACUMULA-1]

- iii) Registrar las recientes lecturas como antiguas:

```
HAZ "PESD1 :NESD1 HAZ "PESD2 :NESD2
```

- iv) Actuar sobre el semáforo:

```
SI (ACUMULA < 200) [M1 "D M2 "P] [M1 "P M2 "D]
```

(Si hay menos de 200 variaciones = 100 coches, se encenderá la luz verde y se apagará la luz roja; en caso contrario, se apagará M1 la luz verde y se encenderá la luz roja;

v) Volver a empezar el ciclo con la llamada recursiva a:

## CICLOCONTROL

El programa definitivo queda como sigue:

PARA INICIAR

```
HAZ "ACUMULA 0 HAZ "PESD1 0 HAZ "PESD2 0
```

```
CICLOCONTROL
```

```
FIN
```

PARA CICLOCONTROL

```
HAZ "NESD1 SD 1 HAZ "NESD2 SD 2 ; paso (i)
```

```
SI NO (:NESD1 = :PESD1) [HAZ "ACUMULA :ACUMULA+1] ; paso (ii)
```

```
SI NO (:NESD2 = :PESD2) [HAZ "ACUMULA :ACUMULA-1] ; paso (ii)
```

```
HAZ "PESD1 :NESD1 HAZ "PESD2 :NESD2 ; paso (iii)
```

```
SI (ACUMULA < 200) [M1 "D M2 "P] [M1 "P n M2 "D] ; paso (iv)
```

```
CICLOCONTROL
```

```
FIN
```

En este ejercicio hemos regulado el sistema mediante sensores atendiendo a sus estados anteriores y a los actuales. Estos sistemas se denominan retroalimentados.

RECUERDA



Los sistemas retroalimentados tienen en cuenta los estados anteriores del sistema.

## VI.- LECTURAS COMENTADAS



## VI.- LECTURAS COMENTADAS

Título: Los muchos rostros de la ciencia

Autor: Antonio Fernández-Rañada

Editorial: Ediciones Nobel

ISBN: 84-87531-44-X

Páginas: 235

Libro que obtuvo el premio internacional de ensayo Jovellanos 1995.

Intenta clarificar las relaciones entre ciencia y tecnología y entre estas dos disciplinas y los demás saberes, mostrando que la ciencia se encuentra íntimamente ligada a los otros saberes humanistas.

Analizando los problemas que actualmente padece la humanidad, propone que las posibles soluciones pasen por la conjunción de las dos culturas - la científica y la humanística.

Libro de debate en el contexto de la tecnología y la sociedad, tema integrado y de suma importancia en el currículo de la educación secundaria.

