

## UNIDAD DIDÁCTICA XIV

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 nueva unidad nos proponemos construir algún pequeño artefacto susceptible de ser controlado mediante el ordenador, utilizar alguna máquina ya construida en unidades anteriores para experimentar con el micromundo de control, y familiarizarnos con un método de trabajo adecuado para tratar con nuestros alumnos contenidos como la comunicación con el ordenador.

## 1.- OBJETIVOS

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

- 1º.- Sistematizar la representación de circuitos.
- 2º.- Diseñar y representar organigramas y algoritmos.
- 3º.- Controlar motores y sensores digitales.
- 4º.- Usar y manipular listas en Logo.
- 5º.- Planificar, contando con las estrategias adecuadas, la introducción al tratamiento de la información por parte del ordenador.
- 6º.- Controlar distintas máquinas

## 2.- CONTENIDOS

En la unidad didáctica que ahora se inicia, habrá ocasión de acercarse a los siguientes contenidos:

### I.- INTRODUCCIÓN

### II.- FUNDAMENTOS DE LOGO: CIFRADO MORSE

### III.-MANOS A LA OBRA

- Símbolos de representación de sensores y actuadores.
- Propuesta de trabajo.
  - La noria
    - Planteamiento del problema
    - Conexiones
    - Resolución del problema

- El puente
  - Planteamiento del problema
  - Conexiones
  - Resolución del problema

Otras propuestas.

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

Estrategias, para plantear en el aula la comunicación con el ordenador: elementos, limitaciones.

- 1.- Elementos de la comunicación
- 2.- El ordenador una máquina con la que nos comunicamos
- 3.- Limitaciones, reglas del juego

#### V.- ENTRE MÁQUINAS Y HERRAMIENTAS

- 1.- Control de un móvil
  - Propuesta de trabajo
  - Conexiones
  - Organigrama y algoritmo
  - Programas
- 2.- Control de un móvil mediante pulsadores
  - Propuesta de trabajo
  - Conexiones
  - Organigrama, algoritmo.
  - Programa.
  - Propuestas

#### VI.- BIBLIOGRAFÍA

##### 3.- CONOCIMIENTOS PREVIOS

Es necesario conocer ya el uso de variables en el lenguaje Logo. Se da por sentado el haber adquirido destrezas para la confección de algoritmos y diagramas; aunque su uso será permanente a lo largo de todas las unidades didácticas y, por tanto, su comprensión podrá afianzarse progresivamente en futuras unidades, es conveniente haber asimilado los contenidos vinculados con variables y algoritmos.

En cuanto a la controladora, deben conocerse perfectamente sus prestaciones y conexiones. El control de motores y sensores digitales debe realizarse con soltura.

## II.- FUNDAMENTOS DE LOGO



## II.- FUNDAMENTOS DE LOGO

### 1.- CIFRADO MORSE.

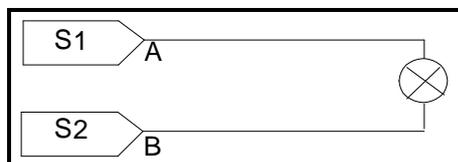
Para repasar y ampliar algunos conocimientos de Win-Logo proponemos un nuevo ejercicio de control sobre una bombilla:

#### 1.a.- EJERCICIO DE CONTROL.



Se trata de realizar el control de una bombilla de modo que: el texto que introduzcamos por medio del teclado del ordenador se visualice en la lamparita cifrado en código Morse.

#### 1.b.- ESQUEMA DE CONEXIÓN DE LA BOMBILLA



Como ya se comentó en esta misma sección de la unidad didáctica anterior, podemos utilizar la primitiva de control M1, que actúa sobre las salidas digitales S1, S2.

Conectando una bombilla de 4,5 voltios a estas salidas, tal como indica la figura, podremos encenderla y apagarla mediante las ordenes: M1 "D y M1 "P.

#### 1.c.- ORGANIGRAMA Y ALGORITMO

##### 1.c.1. Información básica sobre el sistema Morse

Para transmitir un mensaje, el sistema Morse codifica las letras del alfabeto mediante dos signos: "." (punto) y "-" (raya), a los que también se representa por "0" y "1". A cada uno de estos signos corresponde respectivamente un destello luminoso de diferente duración: muy breve para el punto y más largo para la raya. Las letras y palabras se separan con pausas.

Tabla de codificación Morse:

A. _	[0 1]	N_.	[1 0]
B_... .	[1 0 0 0]	O_ _ _	[1 1 1]
C_ . _ .	[1 0 1 0]	P_ . _ .	[0 1 1 0]
D_ . .	[1 0 0]	Q_ . _ . _	[1 1 0 1]
E.	[0]	R_ . _	[0 1 0]
F_ . _ .	[0 0 1 0]	S_ . .	[0 0 0]
G_ _ .	[1 1 0]	T_	[1]
H_ . . .	[0 0 0 0]	U_ . _	[0 0 1]
I. .	[0 0]	V_ . . _	[0 0 0 1]
J_ . _ _	[0 1 1 1]	W_ . _ _	[0 1 1]
K_ . _	[1 0 1]	X_ . . _	[1 0 0 1]
L_ . . .	[0 1 0 0]	Y_ . _ _	[1 0 1 1]
M_ _	[1 1]	Z_ . . .	[1 1 0 0]

1.c.2.- ¿Cómo hacer un punto o raya?

No es difícil hacer que la bombilla emita un destello luminoso equivalente a un punto (o una raya), a la vez que en la ventana de textos se escribe el "." (o "\_") y mientras se oye, además, un pitido breve (o largo) en el altavoz del ordenador; bastan los procedimientos:

#### PARA HAZPUNTO

TECLEA ".\$ La primitiva TECLEA es similar a ESCRIBE, pero no realiza salto de línea al terminar de escribir. El símbolo "\$" indica que el siguiente carácter (un espacio en blanco) se debe escribir.  
Ejemplo: TECLEA "Hola TECLEA "Pepe escribe: PepeHola  
mientras: TECLEA "Hola\$ TECLEA "Pepe da: Pepe Hola.

M1 "D Enciende la bombilla.

TONO 523 5 TONO f n, emite un tono de frecuencia de "f" hertzios durante un tiempo de "n" centésimas de segundo. Mientras suena, el programa sigue corriendo.

ESPERA 5 ; Esta pausa de 5 centésimas hará que el programa espere ese período antes de ejecutar la siguiente línea, la bombilla sigue encendida y se espera a que acabe el pitido. (Duración del destello luminoso correspondiente al punto: 5 centésimas de s).

M1 "P Se apaga la bombilla.

FIN [INTRO]

Análogamente, para hacer una raya "\_" habrá que establecer la duración del destello, que ahora puede ser de 25 centésimas de segundo.

```
PARA HAZRAYA
TECLEA "_$ M1 "D TONO 523 25 ESPERA 25 M1 "P
FIN [INTRO]
```

### 1.c.3.- Datos y variables en Logo

Los tipos principales de datos de Win-Logo son:

Números : 1, 2.53, -7.05,1

Palabras: "Hola, "Bombilla

(precedidas de comillas y sin espacios blancos)

Listas: [1 2.53 Hola [2.53 Hola]]...

(agrupaciones entre corchetes de: números, palabras (sin comillas), o nuevas listas ; separando los elementos con espacios).

Los datos pueden almacenarse en la memoria usando variables; para ello, se utiliza la primitiva: HAZ "nombre\_de\_variable dato.

Ejemplos:

```
HAZ "peso 60
```

```
HAZ "nombre "Mariano
```

```
Haz "alumnos [[Mariano 60][Pilar 50]]
```

### 1.c.4.- ¿Cómo almacenar la tabla de codificación Morse?

Para tener a disposición los datos de la tabla de codificación Morse podemos utilizar la variable "ABECEDARIO en la que, a modo de lista, guardaremos la codificación de cada letra. El primer elemento de la lista será la codificación de la letra "A" (en forma de lista de dígitos: [0 1], equivaliendo el 0 a punto y el 1 a raya). De modo semejante, el segundo elemento representa la "B" y es la lista [1 0 0 0] y así sucesivamente con cada letra del abecedario.

Para disponer de esta variable podremos escribir el siguiente procedimiento:

```
PARA ALFABETO
```

```
HAZ "ABECEDARIO [ [0 1] [1 0 0 0] [1 0 1 0] [1 0 0] [0] [0 0 1 0] [1 1 0] [0 0 0 0] [0 0]
```

```
[0 1 1 1] [1 0 1] [0 1 0 0] [1 1] [1 0] [1 1 1] [0 1 1 0] [1 1 0 1]
```

```
[0 1 0] [0 0 0] [1] [0 0 1] [0 0 0 1] [0 1 1] [1 0 0 1] [1 0 1 1] [1 1 0 0] ]
```

```
FIN [INTRO]
```

### 1.c.5.¿Cómo recuperar un elemento de la lista de codificación?

Para obtener en forma de "devolución" el contenido de una variable se utiliza ":" delante del nombre de la variable.

Ejemplos:

ES 5 + :peso            en la ventana de texto leeríamos 65, si la variable "peso valía 60

Para manejar palabras y listas disponemos de algunas primitivas:

La primitiva PRIMERO (abreviadamente PRI), devuelve el primer elemento de la palabra o lista.

Sintaxis: PRIMERO palabra o PRIMERO lista.

Ejemplos:

ES PRIMERO "manzana            En la ventana de texto se leerá: m  
ES PRIMERO [[a b] c d]            En la ventana de texto se leerá: a b

En el caso de listas, para visualizar los corchetes, es preferible la primitiva MUESTRA a ESCRIBE (ES)

Ejemplos:

MUESTRA PRI [[a b] c d]--->    En la ventana de texto se leerá [a b]  
MUESTRA PRI ABECEDARIO    Se lee [0 1] (codificación de "A")

Si queremos que se devuelva el elemento *n*ésimo de una lista utilizaremos la primitiva ELEMENTO *n* lista.

Ejemplos:

MUESTRA ELEMENTO 3 [a [3 4] [Hola Pedro] 6 [3 Juan]].  
Escribe: [Hola Pedro]

MUESTRA PRI ELEMENTO 3 [a [3 4] [Hola Pedro] 6 [3 Juan]].  
Escribe: Hola

MUESTRA elemento 3 :ABECEDARIO

Escribe la lista de codificación, en Morse, de la tercera letra del alfabeto: "C", es decir, [1 0 1 0]

El ordenador emplea una tabla denominada ASCII (ANSI en Windows) para reconocer los caracteres. En esta tabla, las letras mayúsculas: "A", "B", "C", "D"... ocupan las posiciones correlativas: 65, 66, 67, 68,... Para obtener la posición de un carácter en esa tabla se emplea la primitiva ASCII "carácter.

Ejemplos:

ES ASCII "A            Escribe 65 en la ventana de textos.  
ES (ASCII "A) - 64    Escribe 1

ES (ASCII "B) - 64            Escribe 2

MUESTRA ELEMENTO (ASCII "F) - 64 :ABECEDARIO

- ASCII "F ; devuelve 70, posición de la F en la tabla ASCII
- (ASCII "F) - 64 ; devuelve 6,(70-64), posición de la letra F en el abecedario.
- ELEMENTO (ASCII "F) - 64 :ABECEDARIO devuelve el elemento 6, sexto de la lista de codificación morse "ABECEDARIO, es decir, [0 0 1 0]. Con MUESTRA, se escribirá en la ventana de textos: [0 0 1 0].

#### 1.c.6.- Parámetros en los procedimientos

Al ser llamado el procedimiento siguiente, como sabemos, se dibuja un cuadrado (siempre de lado 50).

```
PARA CUADRADO
REPITE 4 [AV 50 GD 90]
FIN [INTRO]
```

La primitiva AV n, en cambio, hace que la tortuga avance el número de pasos que se especifique en el parámetro n.

Podemos hacer que el procedimiento cuadrado dibuje cuadrados del tamaño que se especifique a través de un parámetro LADO. La definición del procedimiento empieza así: PARA CUADRADO :LADO

Obsérvese que: Se escribe, tras el nombre del procedimiento, un nombre para el parámetro precedido de ":". A partir de ese momento, y sólo dentro de ese procedimiento, podremos utilizar LADO como un variable conocida, que en cada momento tendrá el valor que le asignemos en la llamada a CUADRADO: LADO. El procedimiento completos es:

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

Una vez definido, se comprueba que funciona ejecutando en la ventana de trabajo:

```
BP CUADRADO 25            Borra la pantalla y dibuja un cuadrado de lado 25
BP CUADRADO 50            Borra la pantalla y dibuja un cuadrado de lado 50
HAZ "X 30 CUADRADO :X      Asigna a la variable X el valor 30 y después dibuja
                             un cuadrado de lado igual al valor de esa variable.
```

Recuérdese, en cada caso, de pulsar [INTRO] después de llamar al procedimiento.

### 1.c.7.- Procedimientos Función

Algunos procedimientos entregan valores plenamente operativos en las sentencias. Estos procedimientos se denominan funciones.

Ejemplo: La primitiva AZAR n, devuelve un número natural escogido al azar, entre 0 y el número n-1.

Ejecutar varias veces:

ES AZAR 10 Al hacerlo varias veces, se escribirán en la ventana de textos una serie de valores aleatorios entre 0 y 9.

Podemos construir procedimientos de ese tipo para que nos devuelvan algún valor (número, palabra o lista). Para ello se usa la primitiva DEV valor (Devuelve).

Por ejemplo, si queremos realizar una función que nos devuelva el doble de un número, escribamos el siguiente procedimiento-función:

```
PARA DOBLE :n    "n" representa el parámetro del número del cuyo doble se quiere
                  obtener.
DEV (2 * :n)     Calcula el doble del valor que tenga n y nos lo devuelve.
FIN [INTRO]
```

Téngase en cuenta que al ejecutarse la instrucción DEV valor, el procedimiento terminará con la entrega de dicho valor. Esto quiere decir que: No se ejecutará ninguna otra orden posterior a DEV, que pudiera existir en ese procedimiento.

### 1.c.8.- Obtención, a modo de función, de la codificación morse de un carácter determinado.

Enlazando con lo dicho anteriormente podemos escribir, a tal efecto, el siguiente procedimiento:

```
PARA CODIFICA :LETRA
DEV ELEMENTO(ASCII :LETRA) - 64 :ABECEDARIO
FIN [INTRO]
```

Ejemplos:

En primer lugar, asignaremos a una variable el valor de la letra "F".

HAZ "X "F La variable X toma el valor de la palabra-letra "F"

ES :X        Escribe en la ventana de textos F  
CODIFICA :X Se ejecutará CODIFICA :

Ahora, veamos el procedimiento:

PARA CODIFICA :LETRA La variable X contiene el valor "F", que será traspasado a la variable-parámetro LETRA del procedimiento. Internamente en este procedimiento la variable LETRA valdrá "F"

DEV ELEMENTO(ASCII :LETRA) - 64 :ABECEDARIO  
se transformará en:

DEV ELEMENTO (ASCII "F") - 64 :ABECEDARIO

que por lo explicado en 1.c.5 se transformará en:

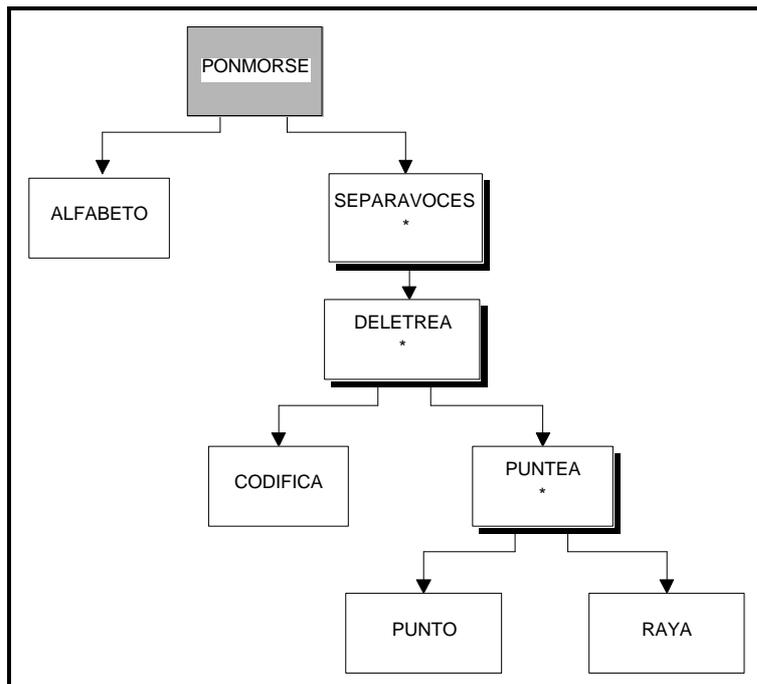
DEV [0 0 1 0]

Con esto terminará el procedimiento entregando la lista [0 0 1 0], que codifica en morse la letra F.

FIN



Si hasta ahora has seguido el desarrollo sin dificultad, adelante. En caso contrario, plantea tus dudas a la tutoría.



### 1.c.9.- Planteamiento global

Ahora que sabemos resolver las cuestiones elementales del problema, veamos el planteamiento global (ver diagrama adjunto) para programar el procedimiento PONMORSE de modo que, al escribir un texto, éste se codifique en MORSE y se visualice en los destellos de la bombilla. Recuérdese que el \* situado debajo del nombre del procedimiento indica que éste es recursivo.

Para poner en Morse (PONMORSE) una frase tendremos que definir el alfabeto Morse, cuestión que ya sabemos resolver con el procedimiento ALFABETO. Escribir una frase en morse equivaldrá a escribir cada una de sus palabras en dicho código haciendo una pausa de separación después de cada palabra.

SEPARAVOCES habrá de separar la lista del texto en palabras, que codificará DELETREA.

DELETREA deberá descomponer cada palabra en letras. Buscará el código MORSE de cada letra mediante CODIFICA y lo enviará, letra a letra, a PUNTEA.

PUNTEA recibirá la lista de "0" y "1" que codifica una letra y, de acuerdo con estos "0" y "1", llamará a HAZPUNTO por cada "0" y llamará a HAZRAYA por cada "1". Los procedimientos HAZPUNTO y HAZRAYA ya están explicados anteriormente en 1.c.2. Con ello se concluirá el trabajo.

#### 1.c.10.- Lectura de listas. Definición del procedimiento PONMORSE

Para leer una lista, desde el teclado, puede emplearse la primitiva LL (Lee Lista)

Ejemplos:

HAZ "X LL	Al ejecutarse, el cursor saltará a la ventana de textos; el texto que allí escribamos se guardará en la variable X, como una lista, en el momento que pulsemos [INTRO]. Si escribimos: Hola María y pulsamos [INTRO], la siguiente orden:
MUESTRA :X	Nos mostrará la lista almacenada en :X, es decir, se leerá en la ventana de textos [Hola María]

El procedimiento PONMORSE puede definirse, de acuerdo con lo anterior, del siguiente modo:

PARA PONMORSE	
ALFABETO	Ya visto, prepara la tabla de codificación en la variable ABECEDARIO
SEPARAVOCES LL	LeeLista, LL, leerá una lista del teclado (el texto que queremos transmitir en Morse) y lo entregará al parámetro de entrada del procedimiento SEPARAVOCES. A este parámetro lo llamaremos ORACION; así pues, dentro de SEPARAVOCES, la variable ORACION será la lista que contiene el texto a transmitir.
FIN	

#### 1.c.11.- SEPARAVOCES. Palabra a palabra

Para transmitir el texto de la lista contenida en ORACION al procedimiento SEPARAVOCES, en primer lugar, mandará transmitir la primera palabra (primero de la lista ; PRI :ORACION); y luego, se llamará a sí mismo, para seguir la tarea con una nueva lista a transmitir (la anterior, una vez eliminada la primera palabra, es decir, MP :ORACION ). Esta llamada recursiva terminará cuando ORACION no tenga más palabras (lista vacía = [ ] )

El procedimiento quedará como sigue:

PARA SEPARAVOCES :ORACION	Oración es la lista que contendrá el texto a transmitir
SI :ORACION=[] [ES ": ALTO]	Ya conocemos el condicional SI y el significado de ALTO. Si la variable ORACION contiene la lista vacía [] el procedimiento terminará y se produce la salida de la recursividad. Previamente, el ordenador escribirá en la ventana de textos ": ", para indicarnos que ha terminado la transmisión del mensaje.
DELETREA PRIMERO :ORACION	Manda al procedimiento DELETREA la primera palabra de ORACION. DELETREA recibirá esta palabra en su parámetro, al que llamaremos VOCABLO
SEPARAVOCES MP :ORACION	Llamada recursiva, se repite la tarea con el resto de la frase (Lista anterior sin el primer elemento)
FIN	

1.c.12.- DELETREA. Letra a letra.

DELETREA: este procedimiento para transmitir la palabra pasada en VOCABLO, letra por letra, recursivamente y de manera análoga al procedimiento anterior, deberá codificar la letra por letra en Morse y mandarla transmitir al procedimiento PUNTEA (que hace los puntos y o rayas).

El procedimiento en cuestión es el siguiente:

PARA DELETREA :VOCABLO

Recibe en VOCABLO la palabra a transmitir

SI (:VOCABLO = " ) [TECLEA " ; \$ ESPERA 30 ALTO]

Se produce la salida de la recursividad si la palabra no tiene letras (:VOCABLO = " ); entonces ALTO terminará el procedimiento (transmisión de esa palabra). En ese caso, previamente, TECLEA " ; \$ escribirá en la pantalla " ; " para indicar el fin de la codificación de una palabra. Además, se hará una pausa de 30 centésimas de segundo para indicar al receptor del mensaje la separación entre palabras (esta pausa se sumará a la pausa de fin de letra que veremos en PUNTEA).

PUNTEA CODIFICA PRIMERO :VOCABLO

PRIMERO :VOCABLO pasará a la función CODIFICA la primera letra de la palabra a transmitir. CODIFICA codificará esa letra, devolviendo la lista de "0" y "1" que le corresponda. PUNTEA recibirá esa lista de "0" y "1", para su transmisión, en su parámetro SIGNOS.

DELETREA (MP :VOCABLO)

Llamada recursiva. Debe continuar haciendo lo mismo con el resto de la palabra (Palabra sin su primera letra MP :VOCABLO)

FIN

1.c.13.- PUNTEA. Transmisión del código de una letra.

El procedimiento PUNTEA recibe en su parámetro SIGNOS la lista de "0" y "1" correspondientes a la letra por transmitir. Por ejemplo, [0 0 1 0] si se trata de una F. Signo a signo, procediendo recursivamente como en los procedimientos anteriores, deberá ejecutar un punto con el procedimiento HAZPUNTO o una raya, con HAZRAYA, según que el signo sea 0 ó 1.

El procedimiento quedará como sigue:

PARA PUNTEA :SIGNOS

SIGNOS recibe la lista de "0" o "1" de la letra por transmitir.

SI :SIGNOS=[] [ESPERA 25 TECLEA " ; \$ ALTO]

Salida de la recursividad, si la lista SIGNOS está vacía. ALTO terminará el procedimiento (transmisión del carácter). Previamente, se escribirá en pantalla " , " (para indicar la separación de carácter). Además, se hará

una pausa de 25 segundos para indicar, al observador de la bombilla, que termina la transmisión de un carácter.

SI PRI :SIGNOS=0 [HAZPUNTO][HAZRAYA]

Se analiza el primer elemento de SIGNOS. Si es un 0, se ordena hacer un punto; en caso contrario (sólo podrá ser un "1") se le manda hacer una raya.

PUNTEA MP :SIGNOS

Puntea debe seguir analizando el resto de elementos de la lista (MP :SIGNOS). Llamada recursiva.

FIN

1.d.- PROGRAMA COMPLETO DE TRANSMISIÓN MORSE A TRAVÉS DE UNA BOMBILLA CONECTADA A LAS SALIDAS DIGITALES S1-S2

El programa completo que se acaba de explicar es el siguiente:

```
PARA PONMORSE
ALFABETO
SEPARAVOCES LL
FIN
```

```
PARA ALFABETO
HAZ "ABECEDARIO [ [0 1] [1 0 0 0] [1 0 1 0] [1 0 0] [0] [0 0 1 0]
[1 1 0] [0 0 0 0] [0 0] [0 1 1 1] [1 0 1] [0 1 0 0] [1 1] [1 0] [1 1 1]
[0 1 1 0] [1 1 0 1] [0 1 0] [0 0 0] [1] [0 0 1] [0 0 0 1] [0 1 1]
[1 0 0 1] [1 0 1 1] [1 1 0 0] ]
FIN
```

```
PARA SEPARAVOCES :ORACION
SI :ORACION=[] [ES ": ALTO]
DELETREA PRIMERO :ORACION
SEPARAVOCES MP :ORACION
FIN
```

```
PARA DELETREA :VOCABLO
SI (:VOCABLO = " ) [TECLEA ";$ ESPERA 30 ALTO]
PUNTEA CODIFICA PRIMERO :VOCABLO
DELETREA (MP :VOCABLO)
FIN
```

```
PARA CODIFICA :LETRA
DEV ELEMENTO(ASCII :LETRA) - 64 :ABECEDARIO
FIN
```

```
PARA PUNTEA :SIGNOS
SI :SIGNOS=[] [ESPERA 25 TECLEA ",$ ALTO]
SI PRI :SIGNOS=0 [HAZPUNTO][HAZRAYA]
PUNTEA MP :SIGNOS
FIN
```

```
PARA HAZPUNTO
TECLEA ",$ M1 "D TONO 523 5 ESPERA 5 M1 "P
FIN
```

```
PARA HAZRAYA
TECLEA "_,$ M1 "D TONO 523 25 ESPERA 25 M1 "P
FIN
```

#### 1.e.- POSIBLES MEJORAS DEL PROGRAMA Y MONTAJE

Puede incorporarse el siguiente conjunto de modificaciones:

- Incluir la "ñ", letras minúsculas, números y símbolos de puntuación en la transmisión.
- Conectar la bombilla a S1 y GND (tierra o masa de la controladora)
- Realizar el montaje decodificador: A través de un sensor lumínico, una célula LDR u otro sensor semejante, leer los destellos cortos, largos y pausas oscuras de separación en la bombilla. Decodificar el sistema Morse traduciéndolo al alfabeto.

## RECUERDA



A lo largo de este ejercicio hemos utilizado la mayor parte de los recursos potentes que nos proporciona LOGO. Debes repasarlos:

- Logo maneja: números, palabras y listas.
- Cómo asignar valores a una variable (HAZ) y como obtenerlos (:)
- Para operar con listas y palabras, entre otras, están las primitivas: PRIMPELEMENTO
- Cómo usar la recursividad para trabajar, uno a uno, con los elementos de una lista.
- Cómo se construyen procedimientos-función (DEV)
- Cómo construir procedimientos con parámetros.
- Uso del condicional SI condición [acciones1][acciones2] para detener la recursividad.
- Algunas primitivas y funciones nuevas, bastante útiles: TECLEA y uso de \$, MUESTRA, AZAR n, TONO f n.



Puedes repasar y ampliar estos conceptos a través del manual de Win-Logo y el apartado II, unidad XII.



### III.- MANOS A LA OBRA



### III.- MANOS A LA OBRA

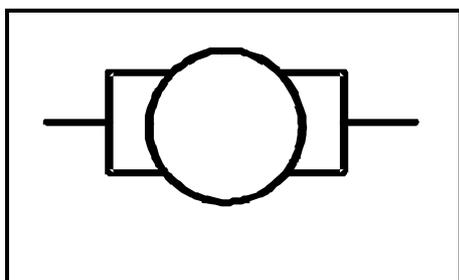
Hasta ahora, de los procesos de robotización, se han tratado fundamentalmente los siguientes contenidos: entradas y salidas, el micromundo de control, los aspectos descriptivos de la controladora y las primitivas del micromundo de control; es este el momento de detenerse en el modo de operar para conectar la máquina por robotizar con la controladora y establecer las características de los sensores y actuadores.

Como una de las características de la tecnología es la disponibilidad de un lenguaje simbólico adecuado para representar correctamente, entre otros sistemas técnicos, circuitos eléctricos y electrónicos, se debe poner el énfasis adecuado en la representación de los operadores usados más habitualmente.

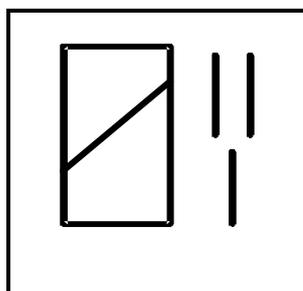


Véase el vídeo de la unidad.

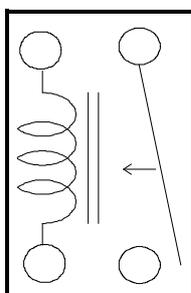
#### 1.-SÍMBOLOS DE REPRESENTACIÓN DE SENSORES Y ACTUADORES.



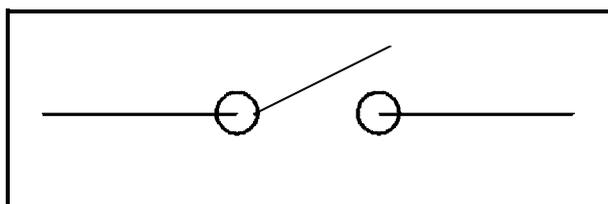
MOTOR



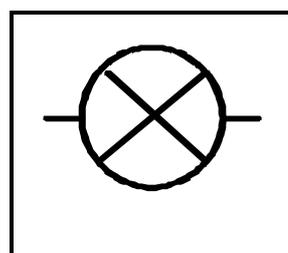
RELÉ



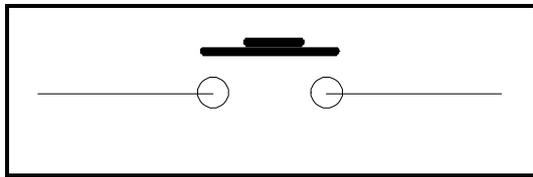
INTERRUPTOR



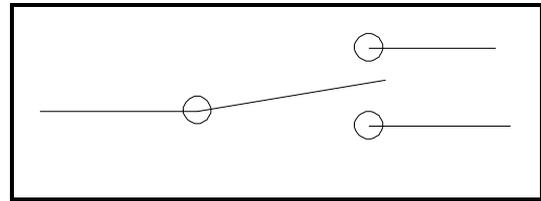
INTERRUPTOR



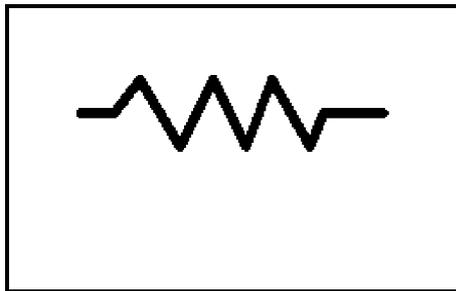
BOMBILLA



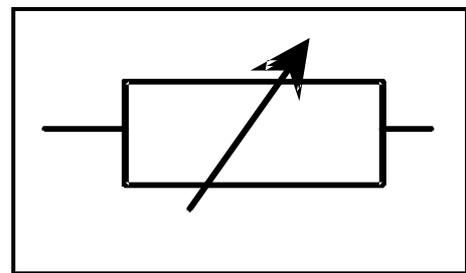
PULSADOR



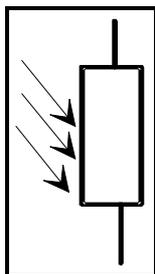
CONMUTADOR



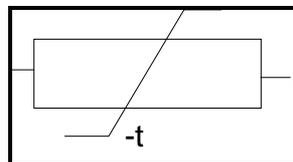
RESISTENCIA



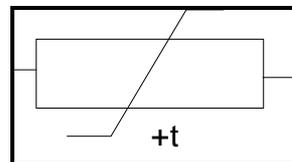
POTENCIÓMETRO



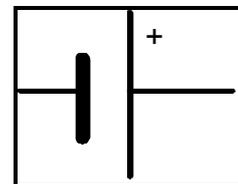
LDR



NTC



PTC



PILA

En esta unidad didáctica, utilizando máquinas ya construidas, analizaremos el proceso de robotización de dichas máquinas, controlando entradas y salidas digitales. Se utilizarán una entrada y cuatro salidas.

## 2.-LA NORIA

### a.-PLANTEAMIENTO DEL PROBLEMA

Ante cualquier problema, como ya se ha dicho en unidades didácticas anteriores, se debe analizar, en primer lugar, la situación por resolver. La propuesta de trabajo consiste en robotizar el funcionamiento de la noria de agua, construida como resultado de otra propuesta de trabajo (Unidad didáctica VI).

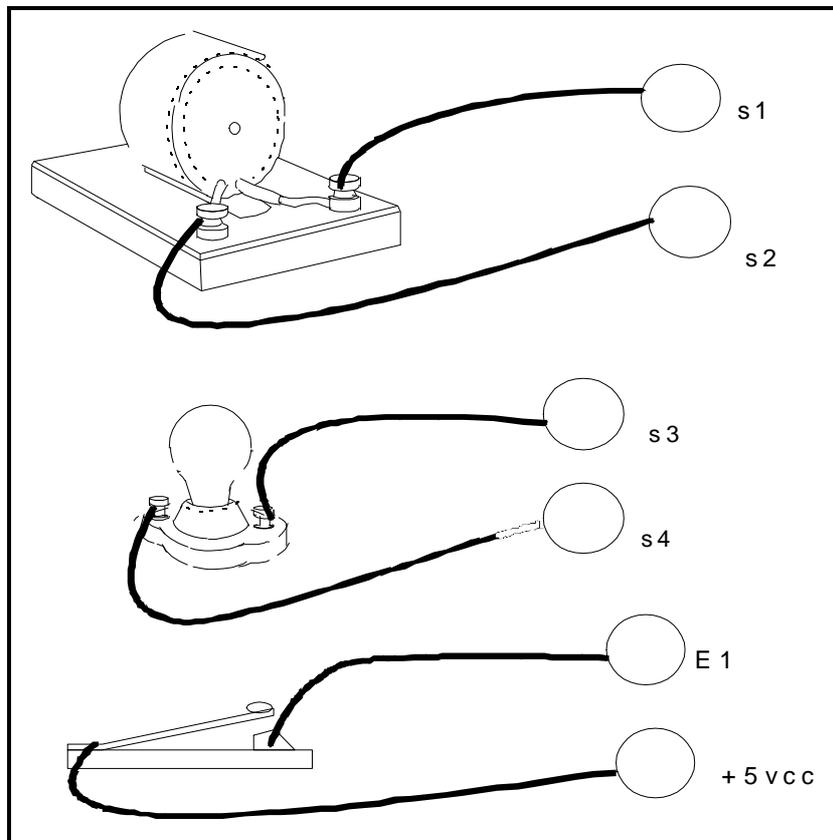
La noria es una máquina bastante simple. Las características de la noria que se pretende controlar son: posee un motor conectado a una polea reductora para hacerla girar, dispone de una bombilla que se encenderá cuando esté en marcha y un pulsador

que servirá para detenerla; la puesta en marcha se hará ejecutando el procedimiento que se desarrollará en lenguaje LOGO. Cuando se construyó la noria, tenía además una pila y un interruptor para la puesta en marcha; ahora estos elementos no son necesarios, pues el motor obtendrá la energía eléctrica de la controladora y la puesta en marcha no se realiza por contactos físicos sino por la ejecución de un programa.

### b.-CONEXIONES

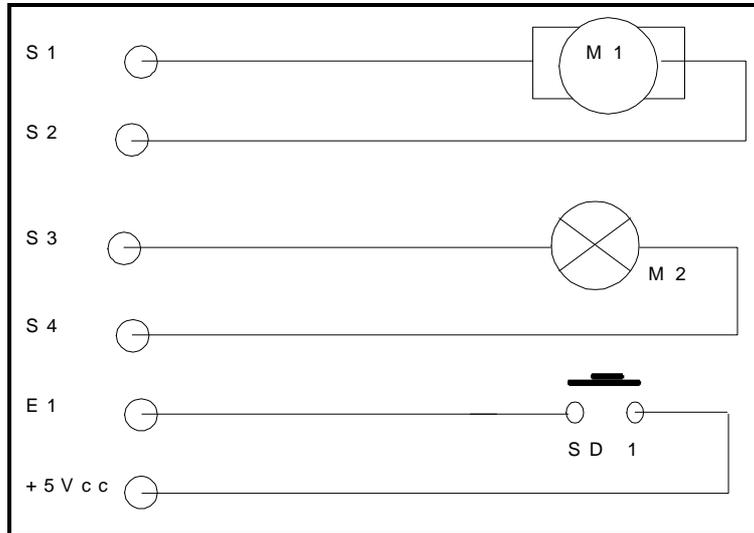
Para el control de la noria se tendrán en cuenta los datos del siguiente cuadro:

SENSORES	ACTUADORES
1 pulsador	1 motor y 1 bombilla
Se conectará a una entrada digital	Se conectarán a cuatro salidas digitales



Los elementos que se deben conectar son los del dibujo, y las conexiones a la controladora serán las siguientes: el motor se conecta a S1-S2; la bombilla, a S3-S4; y el pulsador, a E1 y +5Vcc.

El esquema de conexión de los diferentes elementos, sensores y actuadores, que intervienen en la noria es el que se muestra en el diagrama adjunto. Para una mayor comprensión, se ha señalado el nombre del sensor o actuador en lenguaje LOGO: M1, M2 y SD 1.



### c.- RESOLUCIÓN DEL PROBLEMA

Una vez realizadas las conexiones, debe diseñarse el algoritmo que, trasladado al lenguaje de programación adecuado, resuelva el problema planteado.

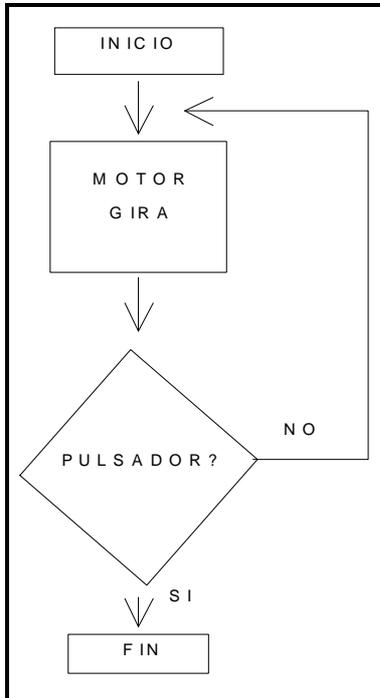
Una primera prueba puede ser la del procedimiento GIRO1. Este procedimiento única y exclusivamente sirve para probar el buen funcionamiento de las conexiones del motor. Después, se tiene que plantear la resolución del problema de una forma más eficaz y más próxima a la realidad.

```

PARA GIRO1
M1 "D ESPERA 200 M1 "P
FIN

```

El algoritmo del proceso de control de la noria puede representarse según el diagrama adjunto. Se inicia el programa y el motor comienza a girar; se plantea una decisión dependiendo del estado del pulsador; según esté o no pulsado se deriva el flujo; en el caso de no estar pulsado, sigue girando el motor y, si lo está, se termina el programa (FIN). El fin del diagrama de flujo del programa tiene que prever la parada del motor antes de terminar la ejecución del programa, pues de no hacerlo cesará la ejecución del programa pero el motor seguirá funcionando.



El procedimiento LOGO puede ser como el que sigue:

PARA GIRO6  
M1 "D

el motor se pone en marcha

SI SD 1 = 1 [M1 "P ALTO ]

si el SD 1 vale 1 el motor se para (M1 "P) y se detiene la ejecución del procedimiento (ALTO)

GIRO6

se inicia el flujo del procedimiento en el caso de no cumplirse la línea anterior.

FIN

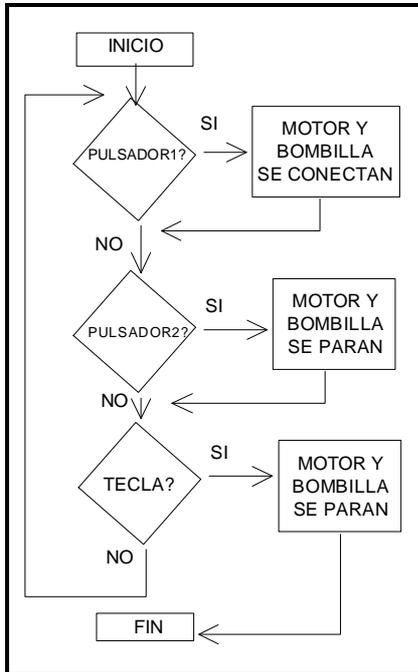
En este primer programa no se ha tenido en cuenta la conexión de la bombilla; el programa puede modificarse para incorporar el encendido y apagado de dicha bombilla:

```

PARA GIRO6
M1 "D
M2 "D
SI SD 1= 1 [M1 "P M2 "P ALTO]
GIRO6
FIN
  
```



Es imprescindible tener construida la noria e ir probando los distintos procedimientos que se muestran. También es conveniente ir realizando variaciones sobre ellos e ir probando distintos tipos de algoritmos.



Una modificación del control de la noria puede ser la de introducir un nuevo pulsador que tendría como misión poner en marcha la noria al ser pulsado. El procedimiento se estaría ejecutando, la noria estará parada y, al pulsarse el primer sensor digital, se pondría en marcha. Al pulsar el segundo sensor se pararía, pero el programa seguiría ejecutándose; para detener la ejecución del programa se debería pulsar una tecla del ordenador.

Siguiendo el algoritmo se puede escribir el procedimiento de la siguiente forma:

```

PARA GIRO7
SI SD 1=1 [M1 "D M2 "D]
SI SD 2=1 [M1 "P M2 "P]
SI TECLA? [M1 "P M2 "P ALTO]
GIRO2
FIN
  
```

### ACTIVIDAD RECOMENDADA



Usando el montaje anterior, diseña un procedimiento en el que la noria gire en un sentido o en otro, según se pulse el sensor uno o el dos. La pulsación de una tecla del ordenador parará la noria y acabará la ejecución del programa.



Envía a la tutoría el procedimiento para su resolución.

### 3.- EL PUENTE

#### a.- PLANTEAMIENTO DEL PROBLEMA

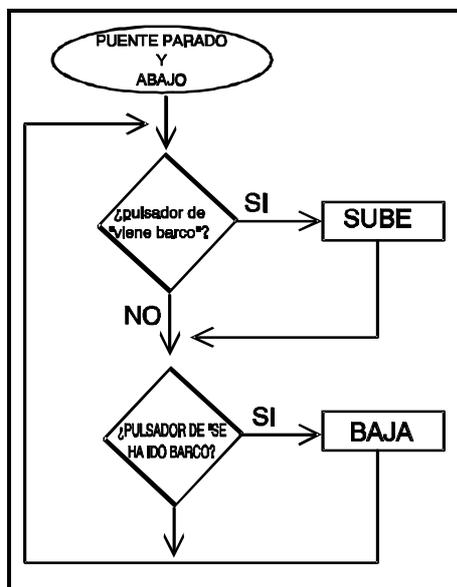
Partiendo de la construcción, ya realizada, del puente levadizo se desea robotizarlo cumpliendo las siguientes condiciones:

- El puente puede subir y bajar.
- Sube cuando se acerca un barco.
- Baja cuando el barco ya ha pasado.

#### b.- LAS CONEXIONES

Para el control de accionamiento del puente se tendrá en cuenta el siguiente cuadro.

SENSORES	ACTUADORES
2 pulsadores	1 motor y 1 bombilla
Se usarán dos entradas digitales	Se usarán cuatro salidas digitales



Se conecta el motor a las salidas S1 y S2 correspondientes controladas por la primitiva M1. Realizadas las conexiones puede diseñarse algún procedimiento de prueba.

En modo directo, puede comprobarse el buen funcionamiento del motor ya conectado.

En la ventana de trabajo se escribirá:

M1 "D ESPERA 100 M1 "P  
el puente se elevará durante 1 segundo.

M1 "D ESPERA 100 M1 "P  
el puente descenderá durante 1 segundo.

Estas órdenes en modo directo pueden superarse mediante un procedimiento que puede escribirse tanto en el editor como en área de trabajo. Los primeros procedimientos pueden ser los siguientes:

PARA GIRO1

M1 "D ESPERA 100 M1 "P  
FIN

PARA GIRO2  
M1 "I ESPERA 100 M1 "P  
FIN

El procedimiento GIRO1 sirve para elevar el tablero del puente, (téngase en cuenta la polaridad), y GIRO2 para bajarlo.

Con el fin de aproximar los nombres de los procedimientos a un lenguaje más natural se pueden crear unos nuevos, cuyo nombre sea más significativo; por ejemplo:

PARA SUBE  
GIRO1  
FIN

PARA BAJA  
GIRO2  
FIN

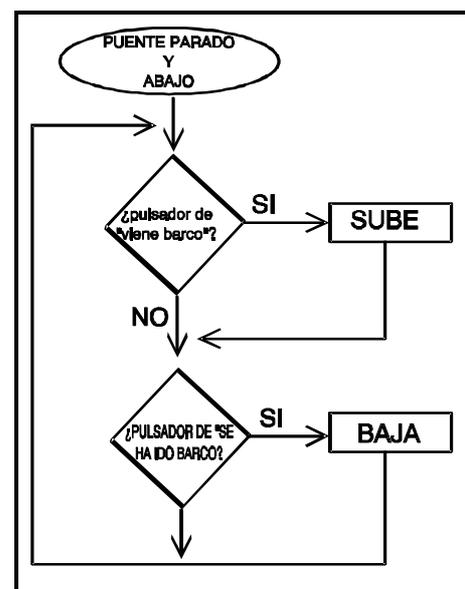
SUBE es sinónimo de GIRO1 y BAJA de GIRO2, pero parece más natural escribir sube y baja.

### c.- RESOLUCIÓN DEL PROBLEMA

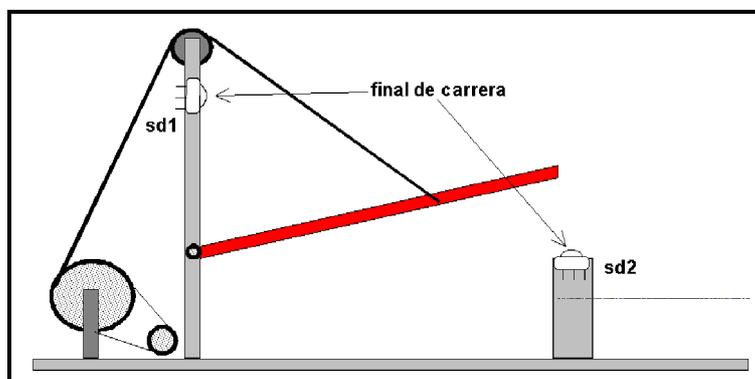
Según los planteamientos iniciales, el algoritmo del problema podría ser el contiguo.

Habrà un pulsador para indicar la llegada de un barco e iniciar la subida del puente. También debe haber un pulsador que indique la salida del barco e inicie la bajada del puente.

Ahora deben instalarse los dos nuevos elementos introducidos; los sensores digitales, dos pulsadores. Uno de ellos se conectará a E1 y +5vcc y el otro, a E2 y también a +5vcc. Se controlan con las primitivas SD 1 el primero y SD 2, el segundo.



La representación gráfica de las conexiones completas es como sigue:



Los pulsadores pueden conectarse en paralelo o de forma individual; en la representación esquemática se han conectado de forma individual. Las nuevas condiciones de la programación del puente son: al pulsar el sensor 1 el puente se eleva y al pulsar el sensor 2, baja. Partiendo del algoritmo mostrado anteriormente se puede desarrollar el siguiente procedimiento que, además, aprovecha los realizados anteriormente (SUBE, BAJA).

```
PARA PUENTE
SI SD 1 =1 [SUBE]
SI SD 2=1 [BAJA]
PUENTE
FIN
```

Escrito el procedimiento y ejecutado, al pulsar el pulsador 1 el puente se eleva y al pulsar el 2, baja. Como no se ha puesto ningún control de abandono del programa, puede hacerse pulsando la tecla [Esc] del ordenador.

#### ACTIVIDAD RECOMENDADA



Modifica el procedimiento anterior para que puede abandonarse el flujo del programa pulsando una tecla cualquiera del teclado.  
Conecta una bombilla que simule un semáforo en rojo para indicar que el puente se encuentra fuera de servicio para los peatones y los vehículos.



Envía el procedimiento que hayas elaborado a la tutoría.

RECUERDA



En la unidad anterior, se describe la primitiva de control que puede activar todos los motores que nos interesen simultáneamente. Puedes probar con ella usándola en los procedimientos que actúen sobre más de un motor.

## IV.- CON NUESTROS ALUMNOS Y ALUMNAS



## IV.- CON NUESTROS ALUMNOS Y ALUMNAS

### 1.- ESTRATEGIAS PARA PLANTEAR EN EL AULA LA COMUNICACIÓN CON EL ORDENADOR.

Es conveniente, antes de comenzar las sesiones en las que trataremos de la comunicación que haya una coordinación con el profesorado de Lengua, Idioma Extranjero, Matemáticas y Educación Visual y Plástica para acordar el enfoque que se da al fenómeno de la comunicación con vistas a presentar un planteamiento interdisciplinar del tema.

Para que se produzca comunicación tienen que darse dos condiciones. Por un lado, dos elementos: el emisor y el receptor. Por otro, un código de comunicación, un lenguaje común que, por medio de normas arbitrarias pero acordadas previamente, les permitirá codificar, enviar y comprender los mensajes que se irán transmitiendo entre ambos.

En la exposición del tema para nuestro alumnado, comenzaremos por todos aquellos elementos de comunicación que en otras áreas del currículum ya han podido ser expuestas y estudiadas. Plantearemos, inicialmente, el análisis del proceso de comunicación que se realiza entre los seres humanos, sin olvidar que los animales también tienen sus sistemas para comunicarse. Por medio de ejemplos sencillos, incluso de juegos, ayudaremos a reflexionar a nuestros alumnos y alumnas sobre las sucesivas fases que se producen en la comunicación humana, haciendo hincapié en aquellos factores que perturban la consecución final del proceso, la comprensión del mensaje por parte del receptor.

Partiremos de la comunicación verbal y comentaremos cómo la idea original, al tener que ser concretada por el emisor por medio del lenguaje, sufre un primer proceso en el que puede quedar algo distorsionada. Una vez decididas las frases, el tono, la inflexión de la voz, acompañarán al mensaje dotándolo de características de intencionalidad, más o menos acordes con la idea original que el emisor intenta transmitir.

Una vez emitido el mensaje, actuarán los medios como el aire, la electricidad o simplemente un hilo que es de esperar no produzcan interferencias que distorsionen la comunicación. El receptor, al captar los sonidos, los traduce a ideas por medio del mismo código, pero no puede evitar introducir valores, su estado de ánimo y hasta sus prejuicios, que condicionarán necesariamente la idea que se forjará del contenido del mensaje que recibe.

Para motivar al alumnado y aportar elementos que le ayuden a analizar los lenguajes y el hecho comunicativo, pueden realizarse las siguientes actividades:

- Proponer el juego del mensaje al oído en el que el contenido primitivo de dicho mensaje, después de pasar por todos los jugadores, queda distorsionado.
- Escenificar situaciones que deben ser contadas quienes están fuera del grupo, para valorar los cambios que introducimos en la comunicación de forma involuntaria.

Para transmitir estos conceptos a nuestro alumnado, nos apoyaremos en todos los conceptos estudiados en las áreas curriculares antes citadas, de modo que sean ellos quienes propongan ejemplos de comunicación oral (directamente, por medio del teléfono, la radio o la televisión), escrita (por medio de carta, telegrama, fax, o la prensa) o simbólica (código de circulación, lenguaje matemático).

En todos los ejemplos que aporten, recalcaremos la necesaria existencia del emisor, el receptor y de un código o norma arbitraria compartida por quienes establecen la comunicación. Otras actividades en las que el alumnado puede analizar los elementos de la comunicación son las siguientes:



- Escenificaciones y juegos donde la comunicación desempeñe un papel importante
- Búsqueda de ejemplos de códigos no lingüísticos como el morse, las banderas, las luces, la circulación, los gestos, el reglamento taurino o el mus.

Aunque la base fundamental de nuestro trabajo en el aula estará centrada en la comunicación entre seres humanos, no tenemos que rechazar tampoco una referencia a los lenguajes usados por los animales. De todos modos, la complejidad de los lenguajes animales en la alimentación, apareamiento y relaciones de dominio territorial son imágenes que muestran un variado abanico que los humanos hemos ido dulcificando o reproduciendo con saludos, gestos, perfumes y demás enmascaramientos civilizados. Una referencia al "Mono desnudo" de Desmond Morris, no estaría fuera de lugar.

Comprendidos los aspectos básicos de la comunicación humana, puede plantearse el tema de la comunicación entre el hombre y la máquina. Durante el desarrollo de la clase, el ordenador y los periféricos estarán presentes y, cuando se haga referencia a ellos, podrán ser utilizados por el profesor o el alumnado como modo de ejemplificación.

La comunicación entre el ser humano y la máquina se produce a través de distintos canales llamados periféricos. Pasaremos a nombrarlos conforme los vayan redescubriendo haciendo notar, en cada caso, su finalidad de introducir información o proporcionarla.

Comprobaremos, utilizando un procesador de texto, cómo se introducen datos mediante el teclado y la pantalla los presenta, cómo con el ratón se introducen órdenes al seleccionar opciones, y la impresora es un periférico por el que el ordenador se comunica con el exterior usando como soporte el papel.

Una vez enumerados y clasificados estos elementos, presentaremos la controladora para ordenador como un periférico por el que el ordenador recibe y, a la vez, envía información al exterior. Mostraremos, por medio de sensores digitales ya conectados, cómo los LED de la controladora se iluminan conforme reciben señal, del mismo modo que podremos ejemplificar como por medio de una orden un motor se pone en marcha, se para o cambia el sentido del giro.

Llegados a este punto, será el momento de introducir a nuestro alumnado en los procesos internos del ordenador, procurando exponer los elementos más fundamentales y buscando ante todo la claridad.

Podemos comentar que en el ordenador se producen comunicaciones internas controladas por un programa que debemos diseñar. El programa estará formado por una secuencia de órdenes escritas en un lenguaje, LOGO, similar al que usamos coloquialmente. Dicho lenguaje tiene su propio vocabulario y su sintaxis las cuales, a diferencia del lenguaje humano, son estrictas, no admiten errores ni interpretaciones.

Las órdenes que crearemos, secuenciadas, formarán el programa, "el pensamiento de la máquina" que se encargará de relacionar toda la información que reciba el ordenador por medio de los distintos canales periféricos con las órdenes de puesta en marcha de los distintos actuadores.

Apoyándonos en láminas y libros de texto sobre anatomía, insistiremos en el paralelismo existente entre el funcionamiento del sistema nervioso humano y la distribución de funciones en una máquina robotizada.

Podemos realizar la siguiente actividad para facilitar la adquisición de conceptos y procedimientos:



- Se divide la clase en dos grupos.
- Un grupo confecciona una lista con los elementos anatómicos que intervienen en un acto voluntario: órganos de los sentidos, nervios sensitivos, cerebro, sensación, producción de una respuesta, nervios motores y músculos que actúan, (aunque sea para dejar al cuerpo quieto y a la espera vigilante).
- El otro grupo confeccionará otra lista con los elementos que intervienen en el control de una máquina: sensores, teclado, ratón, cables de conexión con la unidad central, programa con variables, sentencias condicionales, órdenes, motores que se mueven o están parados.
- Posteriormente, en el conjunto de la clase, se intentará relacionar los elementos de una lista con los de la otra.

A lo largo de la sesión, habremos ido insistiendo en los dos niveles de comunicación que se producen en los procesos de robotización. Por una parte, la máquina recibe información de los sensores y periféricos y da una información a sus actuadores y periféricos. Por otra parte, nosotros estructuramos un programa de control comunicándonos con el ordenador por medio de un lenguaje de programación con una estructura definida efectivamente.

Para desarrollar el primer nivel de comunicación, partiremos de propuestas de ejemplificación aportadas por el alumnado, que traten de temas y actuaciones cercanas a su experiencia: vamos en bicicleta y nos aproximamos a un cruce. Observamos las señales, oímos vehículos y personas, percibimos la velocidad de nuestra bicicleta y el viento, así como el estado del firme. Ante un cambio en las circunstancias de nuestro entorno: un golpe de viento, una luces que nos deslumbran, una pelota que cruza la calle o una mancha de aceite, hacemos actuar los músculos de piernas o manos y pedaleando, frenando o haciendo un determinado movimiento con el cuerpo, intentamos dar una respuesta correcta al estímulo que hemos recibido. Analizando los ejemplos, plantearemos a los alumnos cómo trasponerlos al ámbito de la comunicación propia de los procesos informatizados.

Como toda obra humana, el ordenador, a pesar de ser rapidísima tiene unas limitaciones que vendrán dadas por sus propios elementos constitutivos. Para mejor asimilar las operaciones del procesamiento de la información por ordenador, compararemos sus procesos con las percepciones sensibles.

Partiremos de la especialización de los sentidos, que no nos permite oír con los ojos, ni oler con los oídos; del mismo modo, no todos los sensores captan el mismo tipo de información; por ello, los clasificamos en nuestras máquinas en función del cambio que deseamos apreciar en nuestro programa. Habrá momentos en que varios sensores podrán captar el mismo hecho, del mismo modo que un ciego puede captar la hora por medio del tacto y la presencia de un vehículo por el ruido de su motor. Será entonces cuando deberemos decidir cuál es el sensor que mejor se adapta a nuestro fin, ya sea por fiable, preciso o económico.

Ya hemos planteado antes el paralelismo existente entre la estructura de una máquina controlada por un ordenador y las actuaciones de los seres humanos. Introduciremos la cuestión de las limitaciones del mismo modo haciendo ver, en primer lugar, las limitaciones de nuestros propios sentidos. La comparación con los sentidos de algunos animales deberá ser clarificadora ya que en términos absolutos, todo tiene un límite. Un repaso por los diferentes sentidos permitirá al alumnado valorar las grandes diferencias que caracterizan nuestros sentidos si se comparan con la vista de un búho, o el olfato y el oído de un perro.

Dentro de esta fase de comparaciones, no estaría de más hacer una relación de profesiones en las que la capacidad innata o el desarrollo adquirido por medio del ejercicio, hace que estos profesionales marquen los límites de nuestras percepciones sensoriales. Puede proponerse a la clase que, en pequeños grupos, confeccionen murales sobre los sentidos:



Recoger fotografías o recortes de periódicos donde se plasmen actuaciones profesionales en que los sentidos tienen una gran importancia:

- Catadores de vino, perfume o aceite.
- Profesionales del tinte, la pintura, la fotografía o el vídeo.
- Profesionales de la hostelería: cocineros, pasteleros.
- Relojeros, tratantes de ganado, pastores.
- Peleteros, horteras de comercio textil.
- Músicos y lutiers.

Deberíamos subrayar las limitaciones que supone disponer en el área de tecnología de dispositivos que tienen márgenes de error debido a los materiales empleados para su construcción, las sucesivas utilizaciones y, en suma, el valor en el mercado que está relacionado con los controles de calidad. Sería impensable contar con rayos láser,

discriminadores de voz y demás sensores y actuadores que, aunque están en el mercado, desequilibrarían un presupuesto escolar.

De todos modos, la variedad de los que disponemos abre una amplia gama a la imaginación de nuestro alumnado, que no tiene por qué agotarse durante el desarrollo de los diferentes cursos lectivos. En consecuencia, un profesor o profesora de tecnología no debiera terminar el análisis del tema sin transmitir a su alumnado la idea de que las limitaciones de los elementos periféricos con los que se van a montar las máquinas deben superarse con la inteligencia, la creatividad y la tenacidad del programador. Habrá sobradas ocasiones en las que la falta de un velocímetro podrá ser suplida con un par de sensores y una cinta métrica, ya que el ordenador posee un reloj interno muy preciso y por, si fuera poco, puede emplearse para realizar operaciones matemáticas.



Diseña dos actividades, destinadas al alumnado, para distinguir los elementos que intervienen en el proceso de comunicación. Envía los diseños a la tutoría.

## V.- ENTRE MÁQUINAS Y HERRAMIENTAS



## V.- ENTRE MÁQUINAS Y HERRAMIENTAS

### 1.- CONTROL DE UN MÓVIL

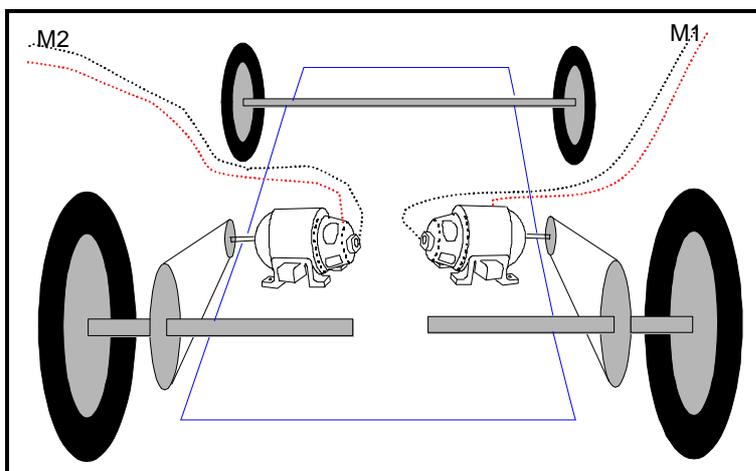
Como sabemos, podemos controlar el giro de un motor en sentido directo o inverso con la controladora para ordenador u otra tarjeta semejante. Suponiendo el motor acoplado a uno de los ejes de un móvil, podremos hacer que el vehículo se mueva hacia adelante o hacia atrás. Así pues, con un solo motor nos limitamos al avance y retroceso del móvil, al control del sentido en una dirección fija.

Para controlar totalmente el móvil es necesario actuar sobre su dirección, de modo que realice giros a izquierda y derecha, además de controlar su avance y retroceso. Es preciso un segundo motor acoplado en cada rueda del tren delantero del móvil. Con ello, cuando actúen los dos motores girando en el mismo sentido, el móvil realizaría un avance o retroceso, y cuando sólo actuara un motor el coche giraría pivotando sobre la rueda parada.

#### 1.1.- CONDICIONES DEL PROCESO DE CONTROL

Se estudiará el control global de un móvil. Hemos de poder actuar sobre el sentido de su marcha: avance-retroceso y sobre su dirección: giros izquierda-derecha. Utilizaremos dos motores, acoplados independientemente, cada uno en una rueda del tren delantero del vehículo.

Suponemos que tenemos el móvil ya construido y los motores acoplados con la correspondiente transmisión, que efectuará la reducción o adecuación de velocidad de rotación de las ruedas.



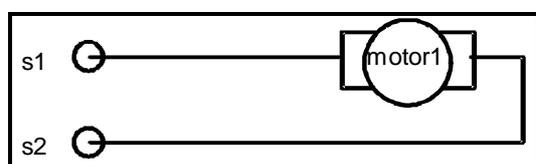
Encendemos el ordenador con la controladora conectada al puerto paralelo (conexión vía LPT1 de la impresora). Enchufamos la controladora a través de su alimentador a la red eléctrica. Entramos en Windows, cargamos Win-Logo (doble "clic" en el correspondiente icono) y cargamos el micromundo de control de la controladora (En la ventana de trabajo ejecutamos:

CARGA "CONTROLA y pulsamos INTRO).

## 1.2.- CONEXIONES

El motor 1 se acopla a la rueda izquierda y el motor 2, a la rueda derecha.

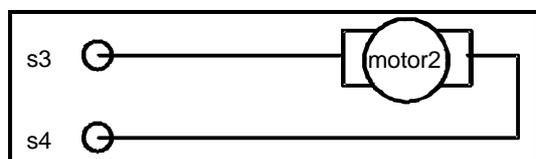
Se conecta el motor 1 a las salidas digitales S1 y S2 y se verifica su funcionamiento. Al ejecutar M1 "D, el motor 1, situado en la rueda izquierda, debe hacerla girar de forma que produzca el avance del vehículo. En caso de que girara en sentido inverso del necesario para dicho fin, se modificará la polaridad de la conexión intercambiando los conectores de S1 y S2. Efectuamos la conexión a nuestra controladora.



M1 "D ha de producir un giro en la rueda izquierda en el sentido de avance del vehículo

Análogamente, se conecta el motor 2 a las salidas digitales S3 y S4 y se verifica su funcionamiento. Al ejecutar M2 "D, el motor 2,

situado en la rueda derecha, debe hacerla girar de forma que produzca el avance del vehículo. En caso de que girara en sentido inverso al necesario, se modificará la polaridad de la conexión intercambiando los conectores de S3 y S4.

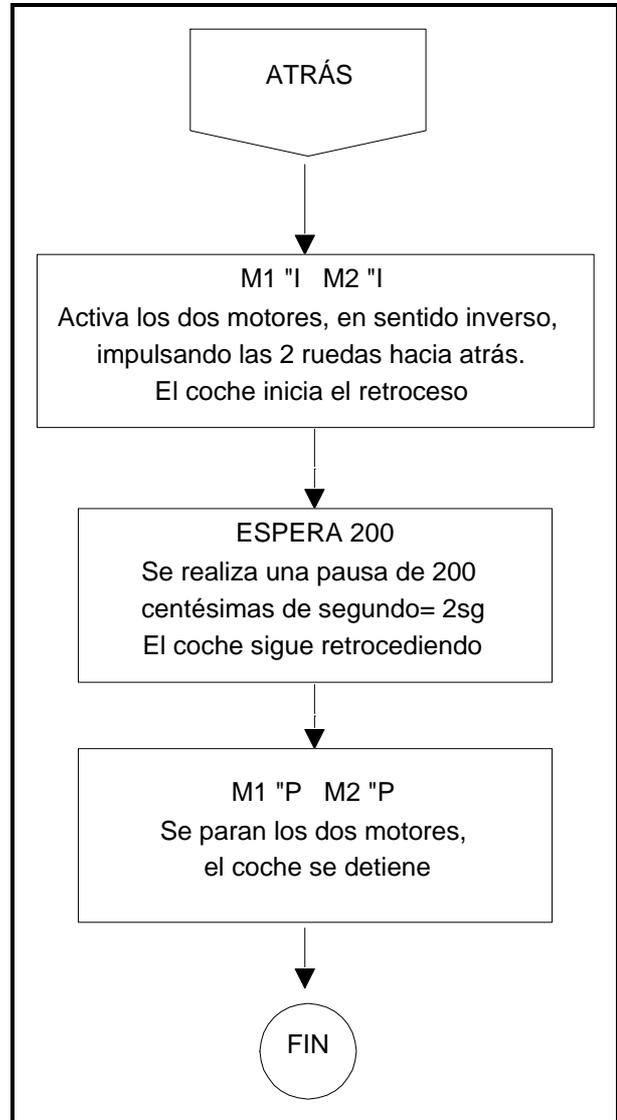
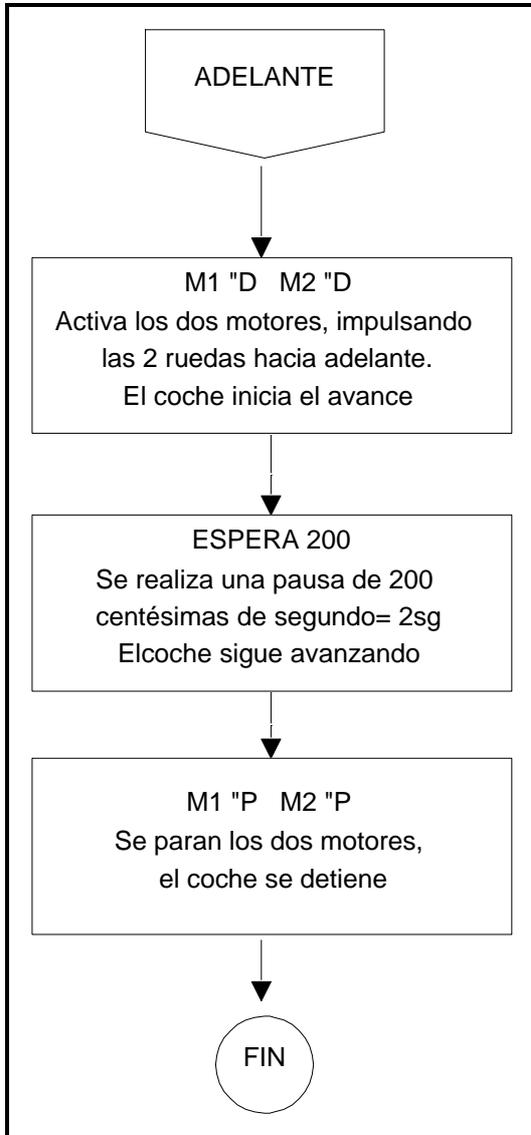


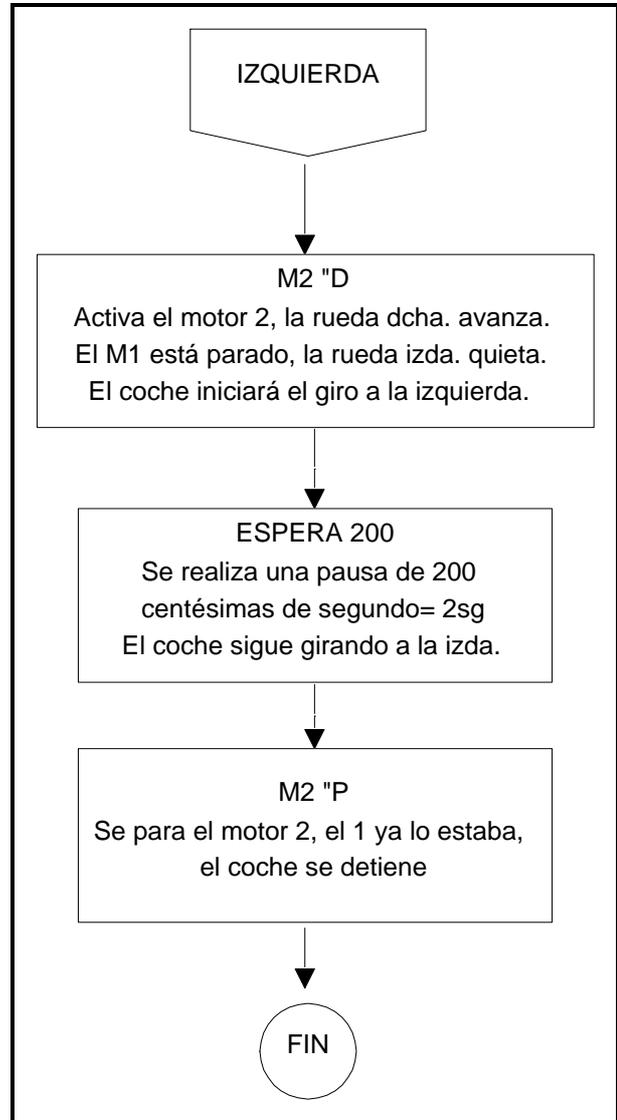
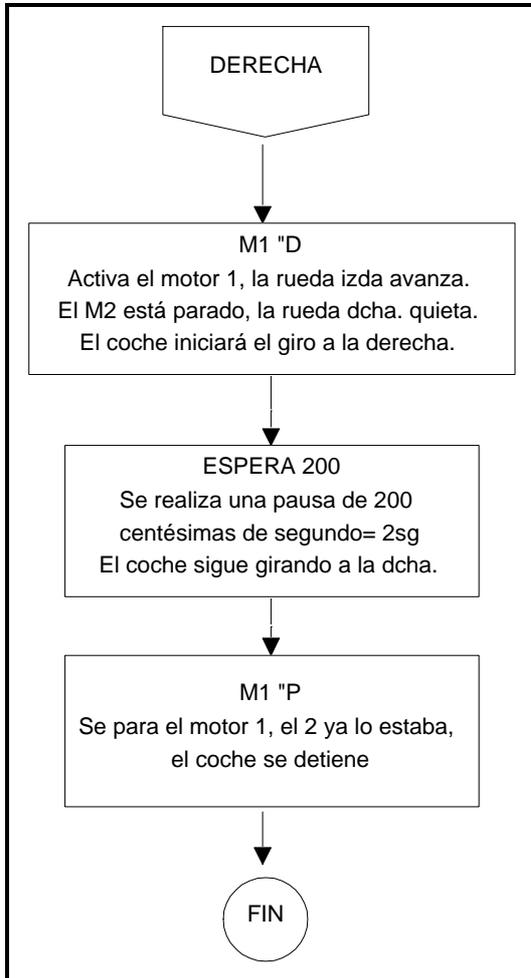
M2 "D ha de producir un giro en la rueda derecha en el sentido de avance del vehículo

Recordamos que M1 "P parará el motor 1 y M2 "P parará el motor 2.

## 1.3.- ORGANIGRAMA Y ALGORITMO

Queremos establecer cuatro procedimientos independientes ADELANTE, ATRÁS, IZQUIERDA Y DERECHA, de manera que realicen los movimientos de avance, retroceso, giro a izquierda y derecha durante un intervalo de 2 segundos.





## 1.4.- PROGRAMAS

Los 4 procedimientos que constituirán el micromundo son:

PARA ADELANTE	PARA ATRÁS	PARA DERECHA	PARA IZQUIERDA
M1 "D M2 "D	M1 "I M2 "I	M1 "D	M2 "D
ESPERA 200	ESPERA 200	ESPERA 200	ESPERA 200
M1 "P M2 "P	M1 "P M2 "P	M1 "P	M2 "P
FIN	FIN	FIN	FIN

Para guardar estos procedimientos agrupados, de manera que constituyan el micromundo de control del vehículo, se siguen los siguientes pasos:

1º Agrupar los cuatro procedimientos en un grupo al que denominaremos COCHE

Una vez escritos los y verificados los cuatro procedimientos, desde la misma ventana de trabajo se ejecuta en modo directo la orden:

AGRUPA "COCHE [ADELANTE ATRÁS DERECHA IZQUIERDA]

La primitiva AGRUPA tiene por sintaxis: AGRUPA "GRUPO [OBJETOS] y realiza el agrupamiento de los procedimientos y variables indicados en objetos con el nombre grupo.



Para acceder a información de ayuda sobre una primitiva, por ejemplo AGRUPA, se escribe el nombre de esta primitiva (AGRUPA) en la ventana de trabajo y teniendo el cursor de escritura sobre ella, desplazamos el puntero del ratón a la barra de iconos, haciendo un doble clic sobre:



2º Para grabar los procedimientos de este grupo se usa la primitiva GUARDA

Se ejecuta a tal efecto:

GUARDA "MÓVIL "COCHE

Con esta orden se habrán guardado en el archivo MÓVIL.LOG (la extensión .LOG la añade por defecto Win-Logo), en el directorio por defecto, los cuatro procedimientos que agrupamos en COCHE: ADELANTE, ATRÁS, IZQUIERDA y DERECHA.

Sintaxis: GUARDA "ARCHIVO "GRUPOoGUARDA "ARCHIVO [OBJETOS]

Cuando en una nueva sesión queramos disponer de esos procedimientos: ADELANTE, ATRÁS, IZQUIERDA y DERECHA, que constituyen el micromundo de control COCHE, bastará cargarlos en la memoria al igual que hacemos con el micromundo CONTROLA de la controladora; para ello, se ejecuta:

CARGA "MÓVILSintaxis: CARGA "Nombre\_de\_Fichero  
(La extensión .LOG se asume por defecto)

A partir de ese instante se dispondrá de los procedimientos grabados en MÓVIL.LOG.

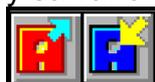


Para más información sobre como cargar y guardar procedimientos se puede recurrir al manual de Win-Logo, al apartado II, unidad XII, o a la ayuda de primitivas del programa Pueden archivar y recuperarse procedimientos, variables, gráficos en distintos formatos o el área de edición completa, con GUARDAR y CARGAR; puede hacerse directamente desde la barra de iconos, pulsando respectivamente en:.

RECUERDA



Para crear, archivar y recuperar un micromundo: AGRUPA los procedimientos, GUARDA y CARGA los mismos. Para cargar y salvar archivos se selecciona el icono



## 2.- CONTROL DEL MÓVIL MEDIANTE PULSADORES

### 2.1.- Mejora del control

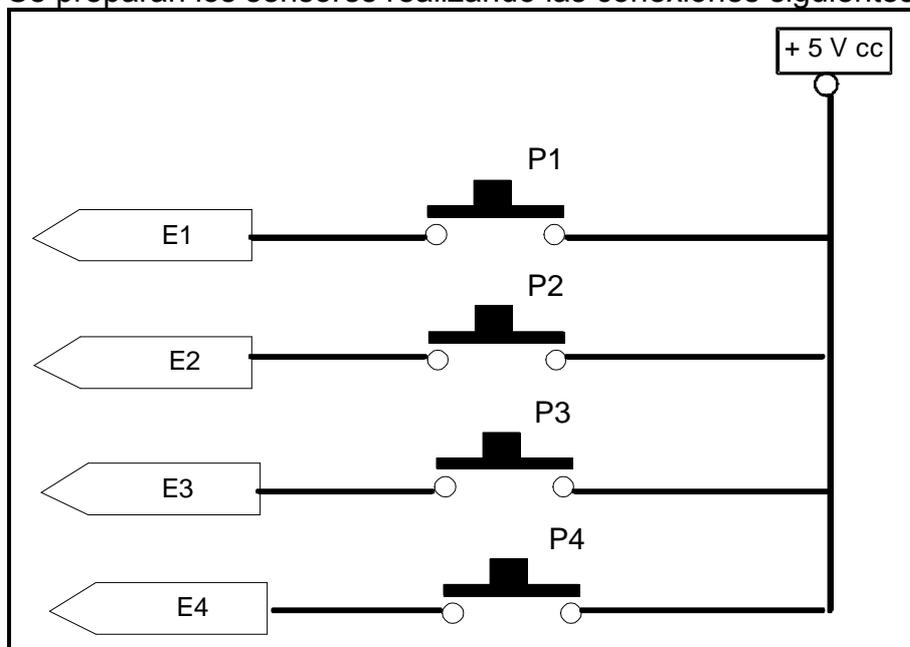
Para hacer más flexible el control del móvil proponemos realizar la conducción del mismo mediante un sistema de pulsadores, de tal manera que se complete un ciclo de realimentación.

Se utiliza un pulsador para cada uno de los cuatro movimientos, que se controlarán mediante cuatro entradas digitales de la controladora: E1, E2, E3 y E4.

Se inicia el proceso con el móvil del apartado anterior, la controladora activada y Win-Logo con los procedimientos de los micromundos CONTROLA y MÓVIL cargados.

## 2.2.- CONEXIONES

Se preparan los sensores realizando las conexiones siguientes:



Un borne del pulsador 1 a la entrada digital E1, el otro a +5Vcc

Un borne del pulsador 2 a la entrada digital E2, el otro a +5Vcc

Un borne del pulsador 3 a la entrada digital E3, el otro a +5Vcc

Un borne del pulsador 4 a la entrada digital E4, el otro a +5Vcc

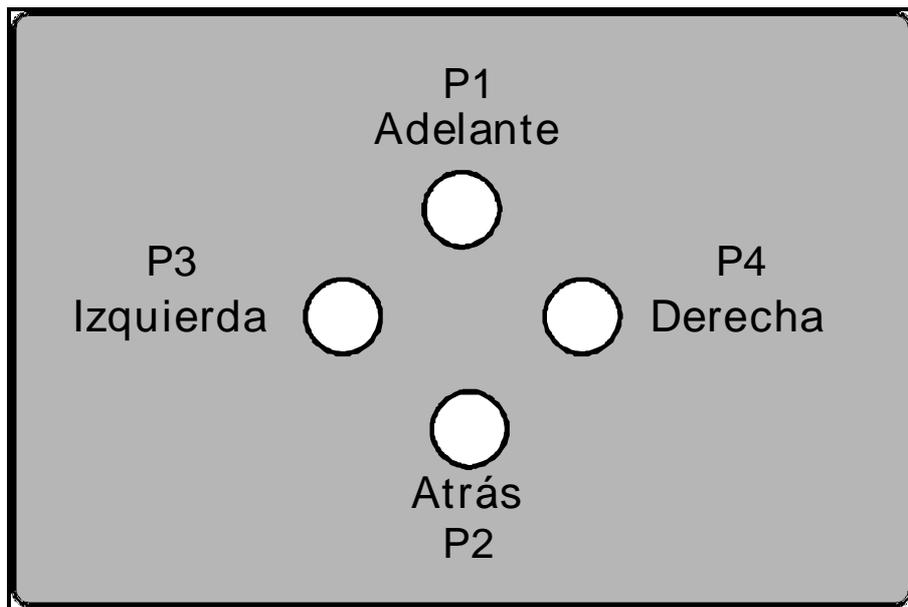
Cuando se cierra el pulsador 1, la tensión que da la controladora en su salida de +5Vcc llegará a la entrada digital E1 ; la lectura de esa entrada mediante la primitiva del micromundo de control SD 1 devolverá un 1. Si no se cierra el pulsador, la entrada E1 estará a 0 Voltios y la primitiva SD 1 devolverá un 0.

Por lo tanto, se tendrá:

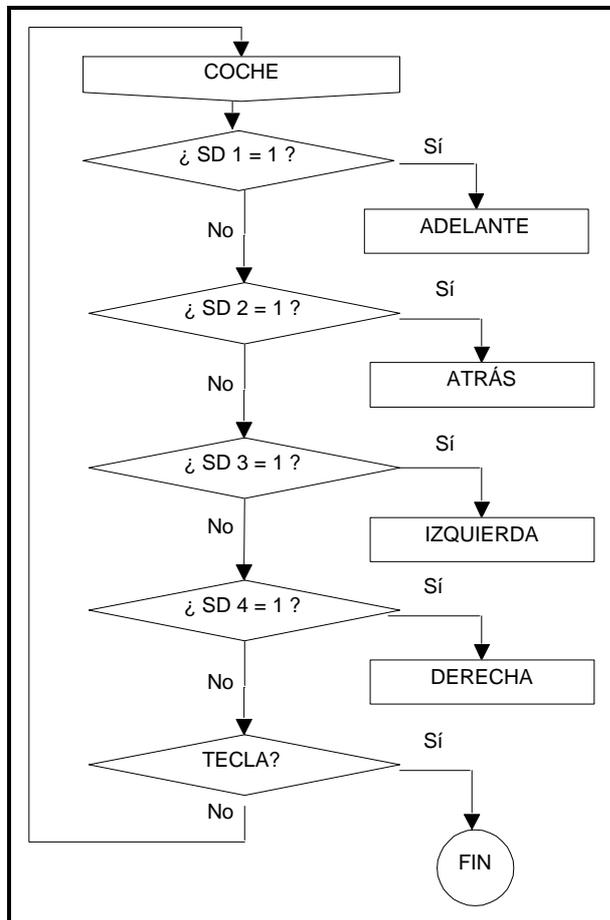
Pulsador y Estado	Entrada y Potencial	Entrada y Estado	Primitiva Devuelve	Acción que pretendemos PROCEDIMIENTO

P1 Abierto	E10Vcc	E10	SD 1---> 0	Ninguna
P1 Pulsado	E15Vcc	E11	SD 1---> 1	móvil: ADELANTE
P2 Abierto	E20Vcc	E20	SD 2---> 0	Ninguna
P2 Pulsado	E25Vcc	E21	SD 2---> 1	móvil: ATRÁS
P3 Abierto	E30Vcc	E30	SD 3---> 0	Ninguna
P3 Pulsado	E35Vcc	E31	SD 3--->1	móvil: IZQUIERDA
P4 Abierto	E40Vcc	E40	SD 4---> 0	Ninguna
P4 Pulsado	E45Vcc	E41	SD 5---> 1	móvil: DERECHA

Conviene acomodar los pulsadores sobre una tablilla que facilite su manejo y colocar en ella los rótulos pertinentes, según se muestra en el siguiente gráfico:



### 2.3.- ORGANIGRAMA, ALGORITMO.



La situación descrita en la tabla anterior facilita la realización del diagrama de flujo adjunto.

Se programará un procedimiento recursivo que realice el ciclo de retroalimentación.

El programa verifica, uno a uno, cada pulsador ¿SD n = 1?, y si está pulsado, llama al procedimiento de movimiento correspondiente que, como sabemos, dura 2 s.

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

### 2.4.- PROGRAMA

El programa de control es ya trivial:

PARA COCHE

SI SD 1 = 1 [ADELANTE];

Si el pulsador 1 está cerrado, el coche avanzará durante 2 s; recuérdese que ADELANTE incluye la orden ESPERA 200. Si en el siguiente ciclo sigue pulsado, el coche continuará avanzando. Si no está cerrado el pulsador 1, el programa prosigue su ejecución.

SI SD 2 = 1 [ATRÁS];

Si el pulsador 2 está cerrado, el coche retrocederá durante 2 s; recuérdese que ATRÁS incluye la orden ESPERA 200. Si en el siguiente

ciclo sigue pulsado, el coche continuará retrocediendo. Si no está cerrado el pulsador 2 el programa prosigue su ejecución.

SI SD 3 = 1 [IZQUIERDA] ;

Si está cerrado el pulsador 3 el coche girará a la izquierda durante 2 s; recuérdese que IZQUIERDA incluye la orden ESPERA 200. Si en el siguiente ciclo sigue pulsado, el móvil continuará girando a la izquierda. Si no está cerrado el pulsador 3 el programa continúa.

SI SD 4 = 1 [DERECHA];

Si está cerrado el pulsador 4 el coche girará a la izquierda durante 2 s; recuérdese que IZQUIERDA incluye la orden ESPERA 200. Si en el siguiente ciclo sigue pulsado, el coche continuará girando a la izquierda. Si no está cerrado el pulsador 4, el programa continúa.

SI TECLA? [ALTO];

Ya conocemos este condicional. Si durante el transcurso del programa pulsamos una tecla, al ejecutarse esta sentencia el programa finalizará. Es evidente que al llegar aquí los motores están parados, ya que cada uno de los procedimientos ADELANTE, ATRÁS, IZQUIERDA Y DERECHA los dejan detenidos al concluir. Si no se ha pulsado una tecla el programa continúa.

COCHE;

Llamada recursiva. Vuelta a empezar de nuevo el programa.

FIN

## 2.5.- PROPUESTAS

Más adelante volveremos con los móviles y trataremos de programar un comportamiento más inteligente: reconocer obstáculos y modificar su comportamiento como consecuencia de ello, obedecer los semáforos o guiarse autónomamente por seguimiento de algún tipo de luz. Planteamos, también, pensar cómo se podría construir y controlar un ascensor de dos pisos.



Consulta a la tutoría si tienes dudas o no has comprendido algún aspecto de la programación.

## VI.- BIBLIOGRAFÍA



## VI.- BIBLIOGRAFÍA

Título: ROBÓTICA  
Colección: Al día en una hora  
Editorial: Anaya Multimedia  
ISBN: 84-7614-727-9  
Páginas:128

Libro de divulgación que, con lenguaje ameno y cordial, introduce al lector en los conceptos y principios básicos de la robótica y su evolución histórica.

Este libro pretende, y lo consigue, poner al día en conceptos sencillos de robótica, tipos de robot, sus aplicaciones, componentes, a la par que posibilita su uso como complemento en el aula.

Su índice indica claramente sus pretensiones y contenidos:

- 1.- Dioses de la Mitología. Introducción a la robótica.
- 2.- De Rómulo y Remo a Rómulo Augústulo. Desarrollo histórico.
- 3.- Patricios, plebeyos y esclavos. Robots, autómatas y simples máquinas.
- 4.- Cristianos en la arena. Descuartizando un robot
- 5.- Las legiones romanas. La fuerza y movimientos del robot.
- 6.- Todos los caminos conducen a Roma. El sistema nervioso
- 7.- Séneca y sus discípulos. El pensamiento del robot.
- 8.- Rosa, rosae. Los lenguajes que pueden hablar.
- 9.- Orgías y bacanales. Los sentidos del robot
- 10.- El incendio de Roma al son de una lira. Ojos y oídos.
- 11.- La vía Augusta. El mercado de los robots.
- 12.- Esclavo para todo. Aplicaciones industriales.
- 13.- Los sirvientes del César. Aplicaciones especializadas.
- 14.- La conquista de Hispania. Las últimas fronteras.
- 15.- Chismorreum imperiali. Noticias, noticias, noticias.

