

# EL PC COMO SISTEMA DE CONTROL



## Introducción

---

El ordenador o PC sigue siendo el entorno de trabajo en todas las industrias y en nuestro hogar. Con ello, se consigue adaptar las tareas que normalmente realiza un operario en una fábrica en una operación controlada automáticamente por medio de un sistema de control mediante el PC y sin la necesidad de operario. En este sentido nos damos cuenta que la capacidad operativa de un ordenador es muy amplia e incluso puede llegar a no tener límites. Pero, para ello, requiere de una tecnología con un alto costo para su implantación. Esta tecnología está vinculada al conjunto de elementos (hardware) que se necesitan: ordenador, sensores, actuadores y fundamentalmente una programación (software) que dirija y ejecute todas las secuencias operativas establecidas por el usuario.

Principalmente, hay que tener en cuenta que el ordenador requiere de una puerta que se abra al exterior, en este caso, para las señales de entradas y salidas, para ello, se instala una **tarjeta de interfaz**, es decir, un medio entre el PC y la máquina, donde se controlan las entradas con **sensores** de captación y las salidas con **actuadores** de accionamiento que actúan directamente sobre la máquina y según la programación establecida.



## Actuadores, sensores y programación

---

La aparición del ordenador en la década de los 40 acelera vertiginosamente el desarrollo de autómatas y robots. La cuestión es: **¿podemos hacer servir el PC como un autómata o un robot?** Para poder responder es preciso verificar si se cumplen las tres condiciones:

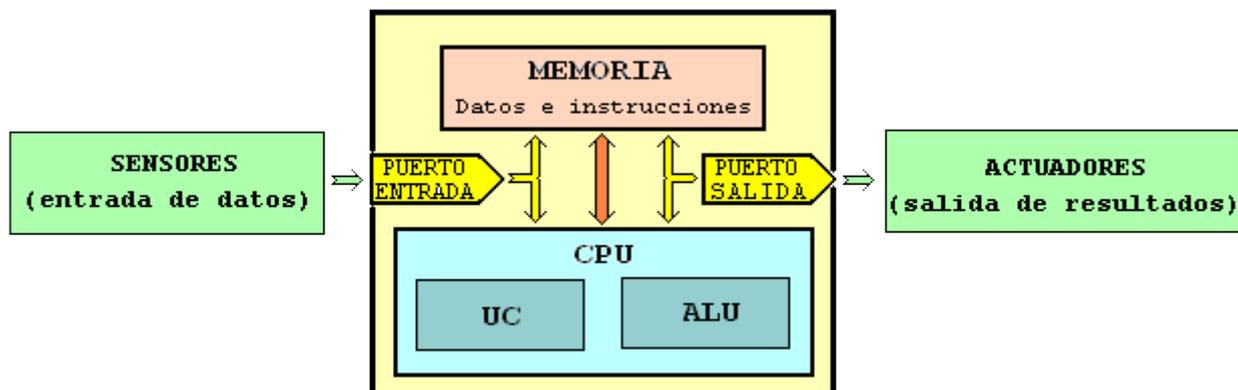
1. ¿Podemos conectarle sensores?
2. ¿Podemos conectarle actuadores?

## EL PC COMO SISTEMA DE CONTROL

3. ¿Podemos programarlo (y reprogramarlo) para que tome decisiones en función de los sensores y de instrucciones previas para que los actuadores operen en consecuencia?

La respuesta a las tres cuestiones anteriores es afirmativa:

- El PC cuenta para comunicarse con sus periféricos, incluso en su versión más básica, con diversos dispositivos de entrada: puertos paralelo y serie, USB, joystick, micrófono... además, es posible agregarle tarjetas especializadas que añaden otras muy diversas clases de entradas.
- También cuenta con varios dispositivos de salida: puertos paralelo y serie, USB, sonido, video... Asimismo, se pueden añadir tarjetas especializadas que expanden el número y tipo de salidas.
- Por otra parte, son muchos los lenguajes de programación utilizables en el PC que permiten leer las entradas y modificar las salidas: BASIC, LOGO, Pascal, C, Ensamblador, etc.



### Conocimiento del entorno: sensores

---

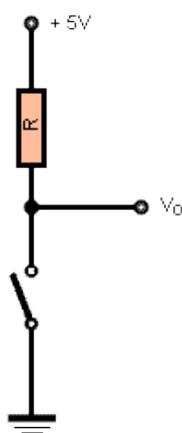
Por medio de ellos el autómatas o el robot conocen la situación del mundo exterior. ¿Cómo recopila esta información?. Supongamos que hablamos por teléfono con un amigo de otra localidad y le preguntamos qué tiempo hace allí. Nos puede contestar de dos maneras posibles:

- Hace sol. Hace calor. El aire es seco.
- La insolación diaria es de 10 h. La temperatura es 26.5 °C. La humedad es del 12%.

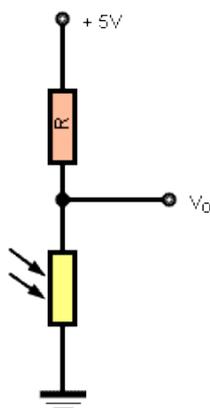
## EL PC COMO SISTEMA DE CONTROL

Las primeras respuestas son de tipo digital: si o no; sol o nubes; calor o frío; sequedad o humedad. Las últimas son de tipo analógico: 10, 9, 5, 2 son posibles valores de las horas de insolación. La temperatura puede oscilar entre  $-10\text{ }^{\circ}\text{C}$  y  $35\text{ }^{\circ}\text{C}$  y, con un termómetro doméstico, se pueden apreciar décimas de  $^{\circ}\text{C}$ . La humedad puede variar entre 0 y 100%.

Las señales digitales se interpretan en el PC utilizando voltajes de 0 V o de +5 V. A una entrada digital se le puede adaptar un elemento de mando del estilo del interruptor de la figura para establecer su valor. Cualquier dispositivo que se comporte de forma similar a éste es útil para el control de entradas digitales.



Muchas señales analógicas se pueden tratar como digitales con circuitos muy sencillos, teniendo en cuenta el cambio que experimenta el sensor utilizado. Véase, por ejemplo, el circuito de la figura. La LDR puede variar de una resistencia del orden de  $1\text{ M}\cdot\cdot$  en oscuridad hasta  $100\cdot\cdot$  a plena luz.



Puesto que la mayoría de los puertos del PC son digitales, otras señales analógicas requerirán, sin embargo, la utilización de circuitos de conversión analógico-digital.

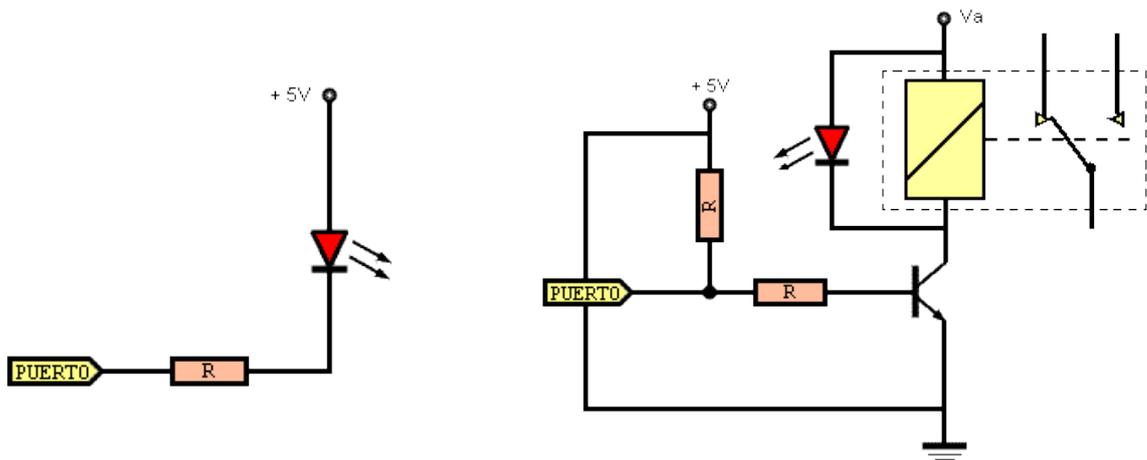
## Modificación del entorno: actuadores

---

Cuando queremos cambiar el entorno podemos hacerlo dando órdenes que pueden ser también de dos tipos: digitales y analógicas. Podemos encender la calefacción, subir una persiana o apagar la luz (digitales), o bien podemos aumentar el volumen de un televisor o regular el termostato de una estufa (analógicas).

Los puertos digitales del PC nos permiten dar órdenes digitales de forma directa o podemos codificar estas señales digitales mediante un conversor analógico-digital para obtener una salida analógica.

Las salidas digitales no proporcionan mucha potencia; como mucho permiten iluminar un LED. Pero se pueden utilizar para excitar un relé o un transistor que controlen el actuador.



En robótica es típico el uso de motores como actuadores. En concreto, por su controlabilidad, los servomotores y los motores paso a paso.

## Dar las instrucciones: programación

---

En general, las máquinas operan a lo largo del tiempo, de modo que el concepto de máquina lleva asociado el de un proceso de funcionamiento en el cual diferentes operaciones se van realizando sucesiva o simultáneamente. Desde el punto de vista del control de su funcionamiento tenemos máquinas no automáticas, o de control manual y máquinas automáticas, que actúan sin necesidad de operador, aunque pueden responder a estímulos externos.

## EL PC COMO SISTEMA DE CONTROL

El funcionamiento de una máquina puede depender únicamente de los elementos que la componen y de sus interconexiones, de modo que ante determinados estímulos siempre responde de manera fija. Pero en otras máquinas automáticas su comportamiento no es siempre el mismo. Estas son las **máquinas programables** y se pueden concebir como una máquina base de comportamiento fijo, que se completa con una parte modificable que describe el funcionamiento de la máquina base. Esta parte modificable se denomina **programa**. Dependiendo de cuál sea el programa que gobierne su funcionamiento, una máquina programable responderá a estímulos externos de una forma u otra. Así se puede comportar como diferentes máquinas particulares en función del programa utilizado. Cuando una máquina opera bajo el control de un programa determinado, se dice que el programa se ejecuta en dicha máquina. Un programa es, por tanto, una **descripción en forma codificada del comportamiento deseado de una máquina**.

**La máquina programable por excelencia es el ordenador**. Éste se define como una máquina programable para tratamiento de la información. Los ordenadores actuales corresponden a un tipo particular de máquina programable que se denominan máquinas de programa almacenado. La estructura general del ordenador es la representada en la figura superior. Como se observa, poseen una memoria donde se pueden almacenar tanto la representación codificada del programa como los datos con los que opera.

La labor de desarrollar programas se denomina, en general, **programación** y los **lenguajes de programación** sirven precisamente para representar programas de manera simbólica, en forma de un texto que puede ser leído con relativa facilidad por una persona (**programa fuente**). Además estos lenguajes son prácticamente independientes de la máquina en la que se van a usar. Pero un programa escrito en un lenguaje de programación simbólico debe ser transformado en el lenguaje particular de cada máquina (**código de máquina**). Los programas en código de máquina son extraordinariamente difíciles de leer por una persona. Normalmente contienen códigos numéricos, sin ningún sentido nemotécnico, y compuestos por millares e incluso millones de operaciones elementales muy sencillas que en conjunto pueden realizar los tratamientos muy complejos que vemos a diario.

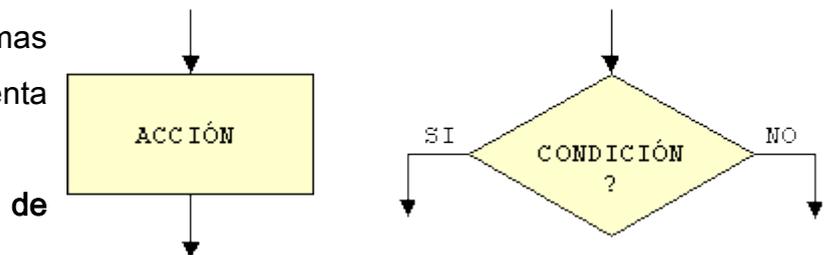
Los mecanismos que permiten ejecutar un programa escrito en un lenguaje de programación simbólico son proporcionados por otros programas denominados **procesadores de lenguajes**: compiladores e intérpretes.

## EL PC COMO SISTEMA DE CONTROL

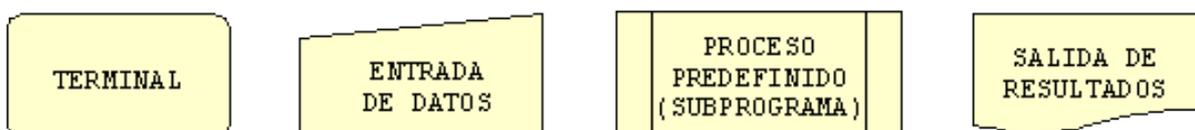
Un **compilador** traduce programas de un lenguaje de programación simbólico a código de máquina. La compilación del programa ha de hacerse sólo una vez, quedando el programa en código de máquina disponible para ser ejecutado en forma inmediata tantas veces como se desee. Un **intérprete** es un programa que analiza directamente la descripción simbólica del programa fuente y ejecuta "sobre la marcha" las operaciones oportunas. El proceso mediante intérprete es más sencillo pero también de ejecución más lenta, ya que hay que ir haciendo el análisis y la interpretación de las operaciones descritas en el programa fuente cada vez que se solicita la ejecución.

El modelo de **programación imperativa** responde a la estructura interna habitual de un ordenador, que se denomina arquitectura Von Neumann. Un programa en lenguaje máquina aparece como **lista de instrucciones** u órdenes elementales que han de ejecutarse una tras otra, en el orden en que aparecen en el programa. Un programa imperativo se plantea como el cálculo o modificación de sucesivos valores intermedios hasta obtener el resultado final. La mayoría de lenguajes de programación actualmente en uso siguen el modelo de programación imperativa.

La estructura de los programas imperativos se representa tradicionalmente mediante **diagramas de flujo** u **ordinogramas** (*flow-chart*).



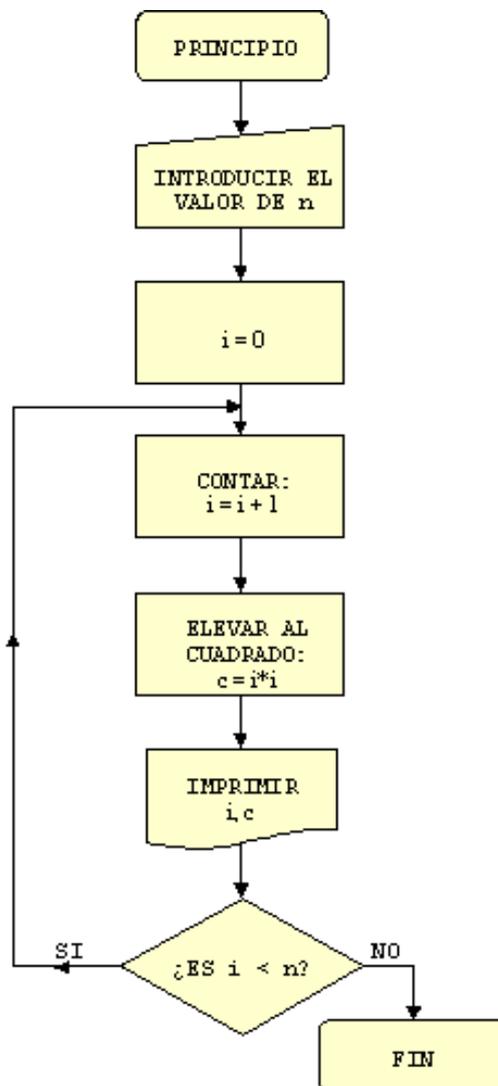
Estos contienen dos elementos básicos, correspondientes a **acciones** y **condiciones**. Existen también otros símbolos convenidos en la representación de ordinogramas, como son:



Utilizando esta representación, podemos esquematizar, por ejemplo, un sencillo programa que imprime una lista de los  $n$  primeros números naturales ( $i = 1, 2, \dots, n$ ) y de sus cuadrados ( $c = i^2$ ). La figura siguiente representa el ordinograma de una estructura secuencial en la que el flujo de control durante la ejecución del programa se refleja mediante líneas o vías que van de un elemento a otro. Las acciones tienen una sola vía de entrada y una de salida mientras que las condiciones tienen una vía de entrada y dos

## EL PC COMO SISTEMA DE CONTROL

de salida. En este diagrama se produce la repetición de ciertas acciones (contar, elevar al cuadrado, imprimir) mientras se cumple una determinada condición (mientras  $i$  es menor o igual que  $n$ ). A esto se le denomina iteración.



Podríamos también haber agrupado las acciones "contar, elevar al cuadrado e imprimir" en un bloque de código designado por un nombre simbólico. A este bloque se le denomina **subprograma** y está el centro de lo que se llama **programación estructurada** (o modular), que no consiste más que en subdividir un problema principal en sub-problemas. Esto resulta ventajoso, no en el sencillo ejemplo descrito, pero sí especialmente en programas que codifican procesos complejos (para reducir la dificultad de la visión global), y en situaciones en las que se repiten operaciones análogas (lo que reduce substancialmente el tamaño y legibilidad del código fuente).

## Una forma de realizar el control programado con el PC

---

Hemos indicado antes que disponemos de una considerable variedad de canales de interacción del PC con el exterior. La mayoría de formas de aprovechar estos canales para realizar el control de dispositivos se puede resumir en dos:

1. Utilizar directamente los puertos de entrada/salida del PC, fabricar la electrónica necesaria para proporcionar potencia -y/o realizar una conversión de señales- y utilizar un lenguaje de programación de propósito general (como BASIC, C o LOGO) para llevar a cabo el control.
2. Utilizar tarjetas de interfaz especializadas comerciales que enlacen directamente los puertos del PC (serie, paralelo, USB) con los dispositivos finales, actuando como "cajas negras" de las cuales sólo interesan sus entradas y salidas. A estas tarjetas se las ha denominado tradicionalmente "controladoras". Entre ellas, las hay de muy diversas marcas y características, y muchos profesores han manejado en los últimos años algún modelo clásico (Inves, Data Harvest, Enconor, ... ) mediante algún lenguaje de programación como LOGO (en alguna de sus distintas versiones). Esta utilización se ha llevado a cabo, en muchos casos, únicamente de forma puntual, dado que no ha existido una dotación regular ni unificada para los distintos centros.

La primera de ambas es muy atractiva pero consume una parte importante del tiempo disponible en la realización de la electrónica mencionada. Además, en algunos casos, requiere una complejidad que supera ampliamente el nivel de conocimientos que se tenga para los que se destina.

Por otro lado, el uso de las interfaces controladoras facilitan enormemente las tareas a la hora de controlar los proyectos que se realizan. Además, en los tres últimos años, las grandes modificaciones que ha sufrido la Tecnología obligan cada vez más inexcusablemente a utilizar recursos para permitir ahorros de tiempo considerables. Desde este punto de vista, existe también otro tipo de interfaces de control comerciales, además de las que clásicamente se han utilizado, orientadas al manejo de conjuntos de control/robótica específicos de algún fabricante, que proporcionan montajes bastante complejos y atractivos y funcionalmente muy aprovechables desde el punto de vista del currículo. Nos referimos a interfaces y conjuntos de control como los proporcionados por fabricantes como FISCHER y LEGO en sus líneas más técnicas y adecuadas (líneas

## EL PC COMO SISTEMA DE CONTROL

"fischertechniks computing" y LEGO Dacta RoboLab). Estos conjuntos tienen un valor añadido nada desdeñable que radica en el lenguaje de programación que incluyen, gráfico y sencillo.

Por tanto, parece conveniente optar por una solución mixta entre el uso de interfaces clásicos orientados al manejo de proyectos que pueda seguir diseñando y construyendo y de interfaces específicos para conjuntos de control/robótica de apariencia y funcionalidad finales relativamente complejos pero de montaje y aprovechamiento relativamente sencillos.

La elección ha consistido en el conjunto de control programado distribuido por la empresa Alecop, constituido por:

- Interfaz de control FlowGo de Data Harvest
- Computer Starter Pack de Fischertechnik
- Software de programación Flowol 2 de Keep I.
- Software de simulación Mimics de Keep I.

y el conjunto de robótica "Fischertechnik computing", constituido por:

- Robots industriales -*Industry robots*-
- Robots neumáticos -*Pneumatic robots*-
- Robots móviles -*Mobile robots II*-
- Interfaz de control -*Intelligent interface*-
- Software de programación LLWin 3

La presente documentación se centra en el aprovechamiento de estos materiales. En la misma, se hace referencia a las interfaces de control de ambos conjuntos y, en cuanto al lenguaje de programación, la documentación se centra inicialmente en Flowol por varios motivos:

- tanto Flowol como LLWin son muy similares en cuanto a concepción y sistemática de uso; ambos son gráficos, con diseños de programa en forma de diagrama de flujo, y con bloques (del diagrama) equivalentes -aunque LLWin cuenta con algunos bloques más y de mayor complejidad-.
- Flowol permite utilizar tanto la interfaz FlowGo como la Intelligent Interface -LLWin sólo se usa con la Intelligent interface-.

## EL PC COMO SISTEMA DE CONTROL

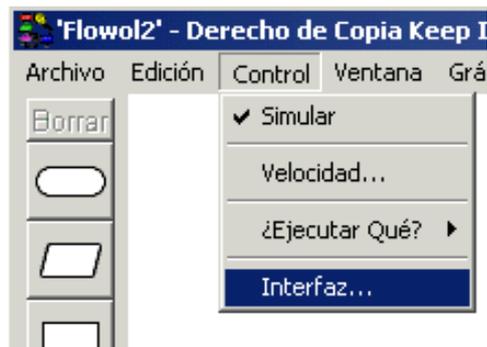
- LLWin cuenta con un sistema de ayuda on-line (zip, 470 KB) dentro de su propio entorno de programación que facilita por sí misma considerablemente la tarea del desarrollo de programas.

### Selección de la interfaz

---

**Flowol** es un lenguaje que permite realizar control programado utilizando diversos interfaces de control ("controladoras") comerciales. Una **interfaz de control es un dispositivo de enlace entre un puerto de salida de un ordenador y los componentes eléctricos/electrónicos de un sistema**. Proporciona potencia a los componentes eléctricos y actúa como *buffer* entre los componentes y el ordenador.

Flowol no está diseñado específicamente para gobernar un único dispositivo. Para acceder a la lista de dispositivos que puede controlar se debe utilizar el submenú **Interfaz** del menú **Control**:



Este menú da paso el cuadro de diálogo de la siguiente figura:

# EL PC COMO SISTEMA DE CONTROL



Como se observa, permite utilizar alrededor de una docena de placas controladoras accesibles mediante puerto serie (Com) y/o USB, entre ellas las de Fischertechnik, LEGO DACTA y FlowGo. El puerto serie que ocupa la interfaz puede seleccionarse mediante los botones de radio **Puerto de comunicaciones** y **Com X**.

Un ejemplo de conexión al puerto USB se puede observar a continuación en el caso de la interfaz **FlowGo**. Cuando se introduce el conector del cable USB adjunto a la interfaz en el puerto USB del ordenador se produce la detección automática del dispositivo y la instalación del archivo controlador de dispositivo correspondiente (caso de Windows a partir de 2000/XP):



# EL PC COMO SISTEMA DE CONTROL

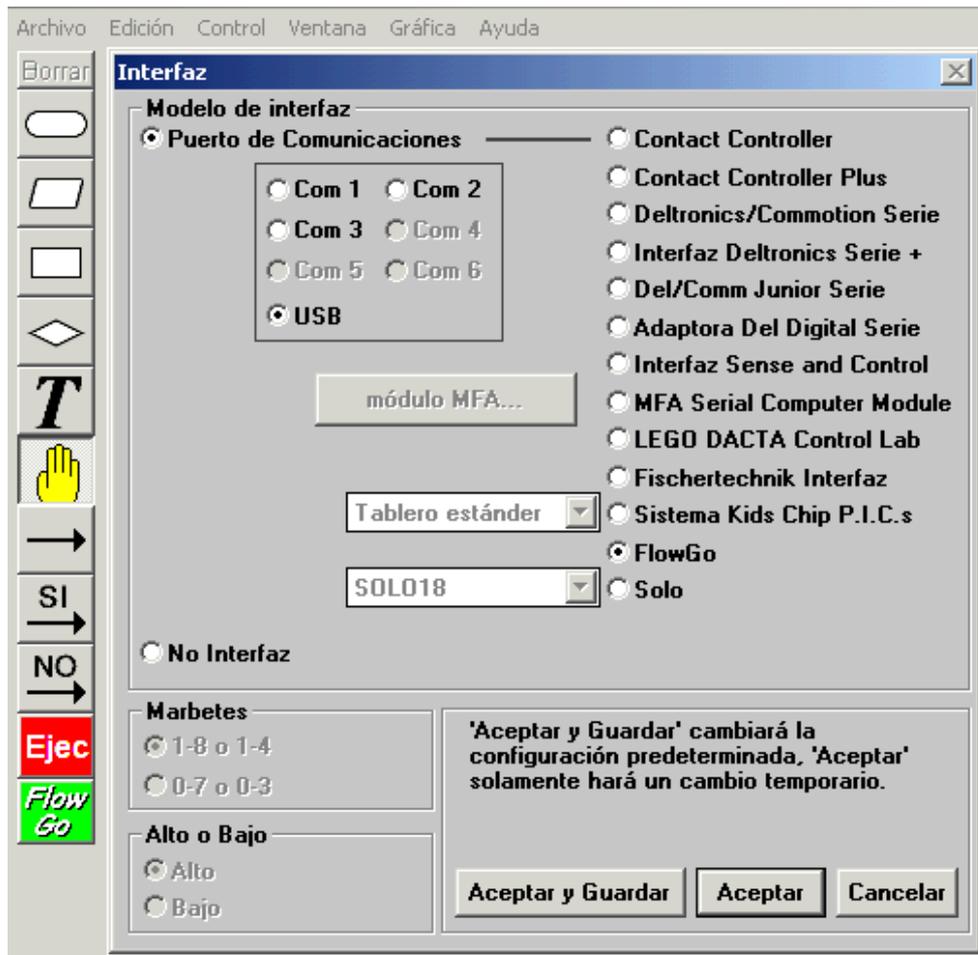


Cuando se finaliza la instalación hay que reiniciar el equipo y, tras conectar la interfaz, si el sistema no busca automáticamente el **driver** adecuado proporcionado por la instalación anterior, se puede acudir a "Panel de control > Administrador de dispositivos" y en el "Dispositivo Desconocido" denominado "FlowGo/Solo Device" obligar a actualizar el controlador de forma que en la búsqueda encuentra automáticamente el *driver* correcto.

Cuando el sistema finaliza la detección y la instalación del *driver* de dispositivo adecuado, ya se puede entrar en Flowol y seleccionar el puerto USB para la utilización de la interfaz.

En la siguiente figura se muestra la interfaz FlowGo con los puertos de comunicaciones disponibles:

# EL PC COMO SISTEMA DE CONTROL



Una vez seleccionada la interfaz de control de la que disponemos y el puerto correcto mediante el cual se comunica con el PC, podemos guardar la configuración permanentemente mediante el botón **Aceptar y Guardar**.

Elegida la interfaz, las pantallas de **monitor reducido** (submenú **Monitor Reducido...** del menú **Ventana**):



y **monitor amplio** (submenú **Monitor Amplio...** del menú **Ventana**):

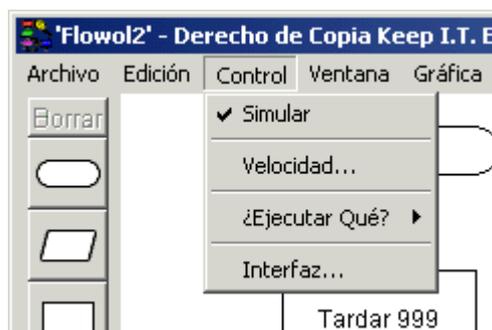
## EL PC COMO SISTEMA DE CONTROL



permiten observar a tiempo real el estado de las entradas y salidas de la interfaz (el nº y características de las entradas/salidas que se muestran depende del tipo de interfaz que se haya seleccionado).

Las etiquetas que se observan (*Val*, *Entrada*, *Salida*, *Motor*) son modificables por el usuario para indicar qué tipo de sensor o actuador se halla en cada entrada o salida, respectivamente.

En la fase de desarrollo de un programa no es preciso tener conectada la interfaz, de modo que se puede poner el sistema en modo **Simular** (marcando la opción correspondiente del menú **Control**).



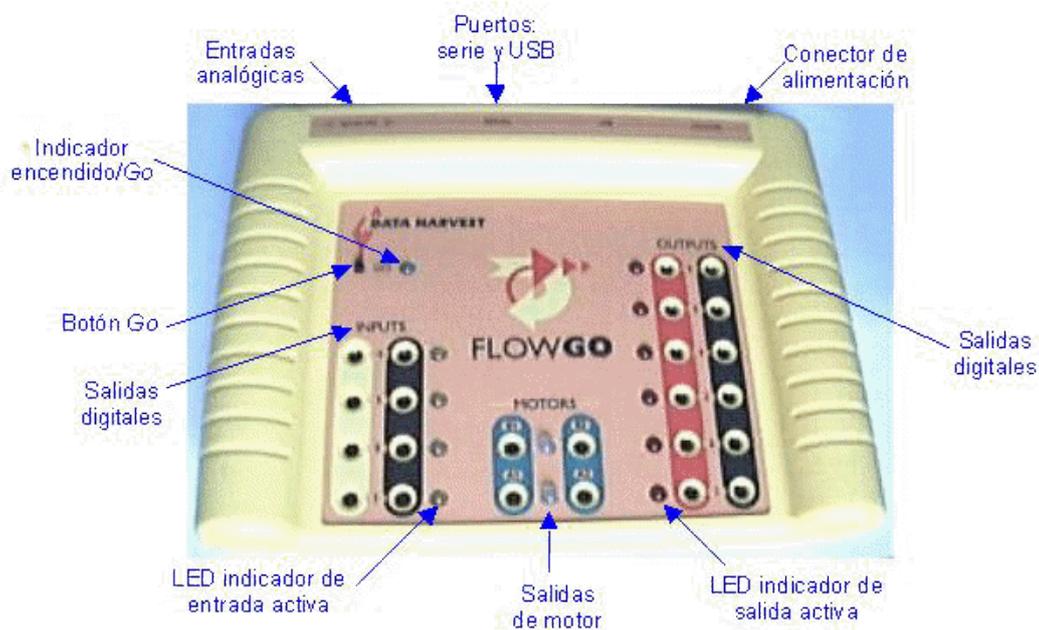
Esto permite diseñar, probar y depurar un programa sin necesidad de la conexión física al interfaz.

Con **Flowol** utilizaremos principalmente las interfaces de control **FlowGo** de Data Harvest e Intelligent interface de fischertechnik por su sencillez, versatilidad y equipamiento asociado.

# EL PC COMO SISTEMA DE CONTROL

## La interfaz de control FlowGo

Nos referiremos ahora a la interfaz FlowGo, que se muestra en la figura siguiente:



Sus características principales son:

<b>Alimentación</b>	Requiere fuente de alimentación con salida de <b>6V cc. y 2 A</b> o 4 pilas tipo 'C' o 'D' de 1.5 V (en porta-baterías con conector estándar) Corriente nominal proporcionada a las entradas/salidas: 1.3 A
<b>Puerto de control</b>	1 <b>puerto serie</b> RS 232, conector DB 9 1 <b>puerto USB</b> (con conectores y cables para ambos puertos disponibles)
<b>Salidas</b>	<b>6 digitales</b> de 6 V, 100 mA <b>2 de motor</b> , con inversión y regulación de velocidad, 200 mA (modulación de anchura de pulso a 62 Hz).
<b>Entradas</b>	<b>4 digitales</b> (sensores on/off) ( $R_{ONmax} = 4K$ , $R_{OFFmin} = 5K5$ ) <b>2 analógicas</b> (sensores de nivel) (Voltaje de referencia: 2.5 V, 1 mA por canal)
<b>Indicadores</b>	6 LED indicadores de salida digital 2 LED indicadores de motor 4 LED indicadores de entrada digital 1 LED indicador de encendido y/o de ejecución de programa memorizado
<b>Botones</b>	Botón <i>Go</i> para iniciar la ejecución de un programa memorizado
<b>Memoria</b>	256 bytes (equivalente al almacenamiento de 100-200 símbolos de Flowol)

## EL PC COMO SISTEMA DE CONTROL

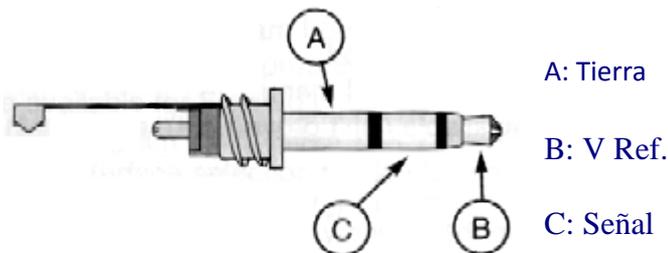
Las clavijas de las entradas/salidas digitales y de motor son del tipo "banana de 4 mm". Para conectar un dispositivo como una lámpara, un LED (protegido por una resistencia de 470 W), un zumbador o un motor de c.c. no hay más que conectar los dos polos en las entradas correspondientes:



Análogamente para las salidas de motor:



Los conectores de las entradas analógicas son del tipo "jack estéreo de 3.5 mm", con tres patillas descritas como en la siguiente figura:



Por ejemplo, para medir la resistencia de un potenciómetro lineal de 10 K, se debe conectar la patilla deslizante (la central) a la Señal (C) y las dos patillas fijas (las de los extremos) a Tierra (A) y V Ref. (B).

Cuando se lea la entrada del sensor, se obtendrá una lectura 100% cuando la resistencia del potenciómetro sea 10 K, y una lectura 0% cuando la resistencia sea 0 K.

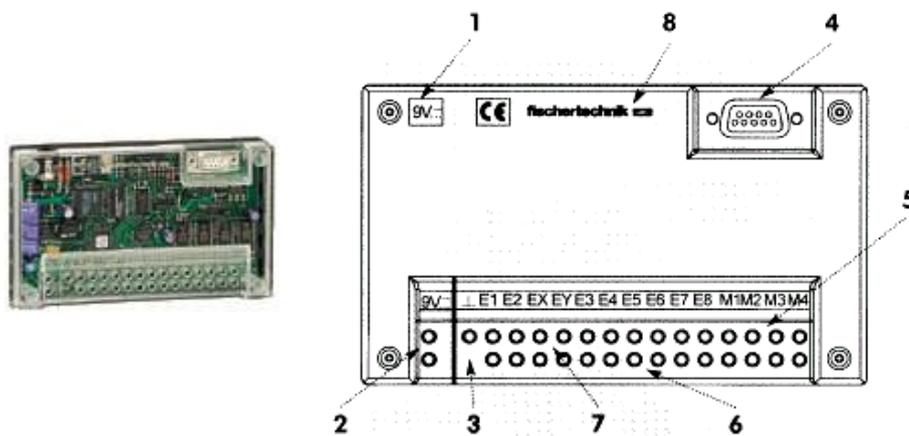
# EL PC COMO SISTEMA DE CONTROL

La interfaz puede funcionar en dos modos:

- **modo en-línea** ("on-line", gobernada en todo momento mediante conexión al PC), ó
- **modo descarga**: en este modo el procesamiento del programa tiene lugar únicamente en el microprocesador de la interface, desacoplado del PC. En una primera fase se diseña el programa de control en el PC, luego se descarga en la interfaz y, finalmente, ésta puede cortar el contacto con el PC (de modo que el cable serie o USB puede ser desconectado).

## La *intelligent interface de fischertechnik*

Nos referiremos ahora a la "interfaz inteligente" e fischertechnik, que se muestra en la figura siguiente:



Sus características principales son:

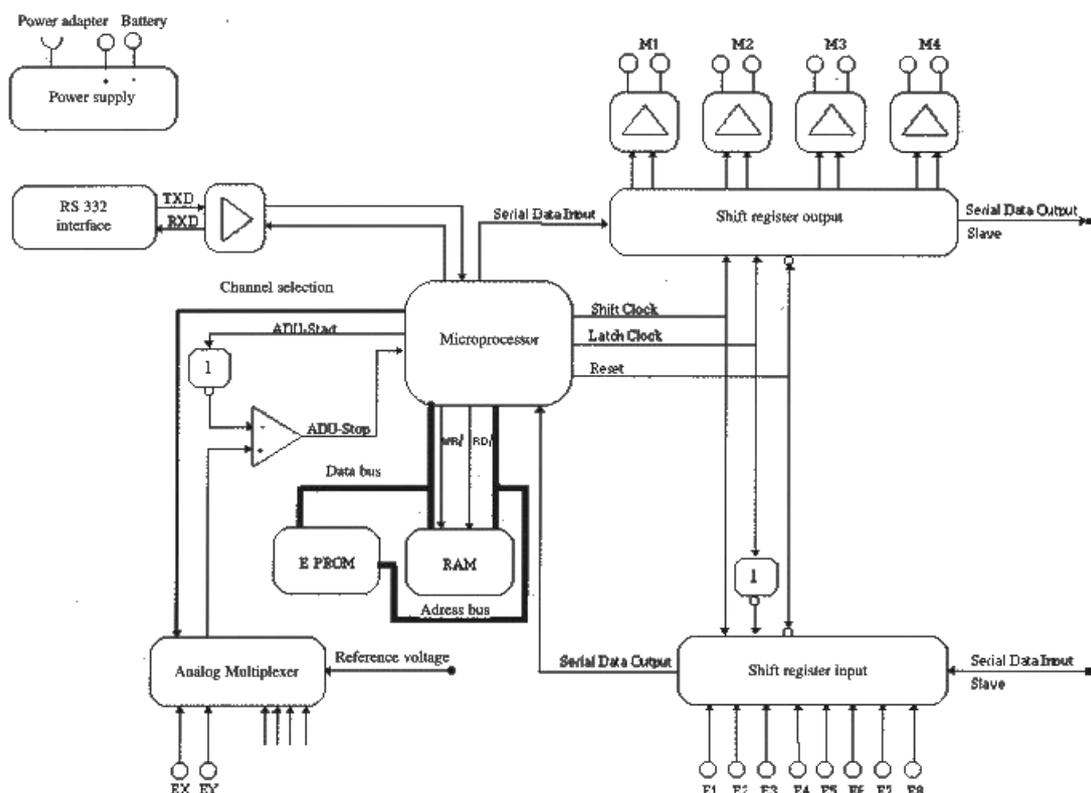
<b>Alimentación</b>	Requiere fuente de alimentación con salida de <b>9V cc. y 1 A</b> , Energy Set 30182 de fischertechnik -o cualquier alimentador universal de dichas características <b>con el polo + en el interior del casquillo-</b> (conexión en 1, ver la figura); o 1 pila de 9 V, o batería recargable "Accu Set" 34969 de fischertechnik para uso con robots móviles (Mobile Robot II) (conexión en los casquillos 2 de la figura) Limitador de corriente a 1A, resistente a cortos.
<b>Puerto de control</b>	1 <b>puerto serie</b> RS 232, conector DB 9 (ver 4 en la figura)
<b>Salidas</b>	<b>4 digitales/de motor</b> (con inversión pero sin regulación de velocidad), 9V, 250 mA M1, M2, M3, M4 (ver 5 en la fig.)

## EL PC COMO SISTEMA DE CONTROL

<b>Entradas</b>	<p><b>8 digitales</b> (sensores on/off), E1 - E8, 6 - 12 V, <math>R_{\text{entrada}} = 9 \text{ K}</math> aprox., nivel de conmutación = 2.8 V (ver 6 en la fig.)</p> <p><b>2 analógicas</b> (sensores de nivel), EX, EY, <math>R = 0-5 \text{ K}</math>, niveles = 0-1023, tolerancia = 0.2% (ver 7 en la fig.)</p>
<b>Indicadores</b>	1 LED rojo indicador de encendido
<b>Botones</b>	Ninguno
<b>Memoria</b>	<p>RAM de 32 kByte (para almacenamiento de un programa; si se interrumpe la alimentación se pierden los datos del mismo).</p> <p>EPROM de 64 kByte (sólo para el <i>software</i> del microprocesador)</p>
<b>Ampliación</b>	Módulo de extensión para duplicar entradas/salidas digitales conectado mediante regleta de 14 polos (ver 8 en la fig.)

Las clavijas de las entradas/salidas son de 2 mm, proporcionadas en los diferentes modelos de montaje de la línea *fischertechnik computing*.

En la siguiente figura se puede observar un diagrama de bloques de la composición interna de la interfaz:



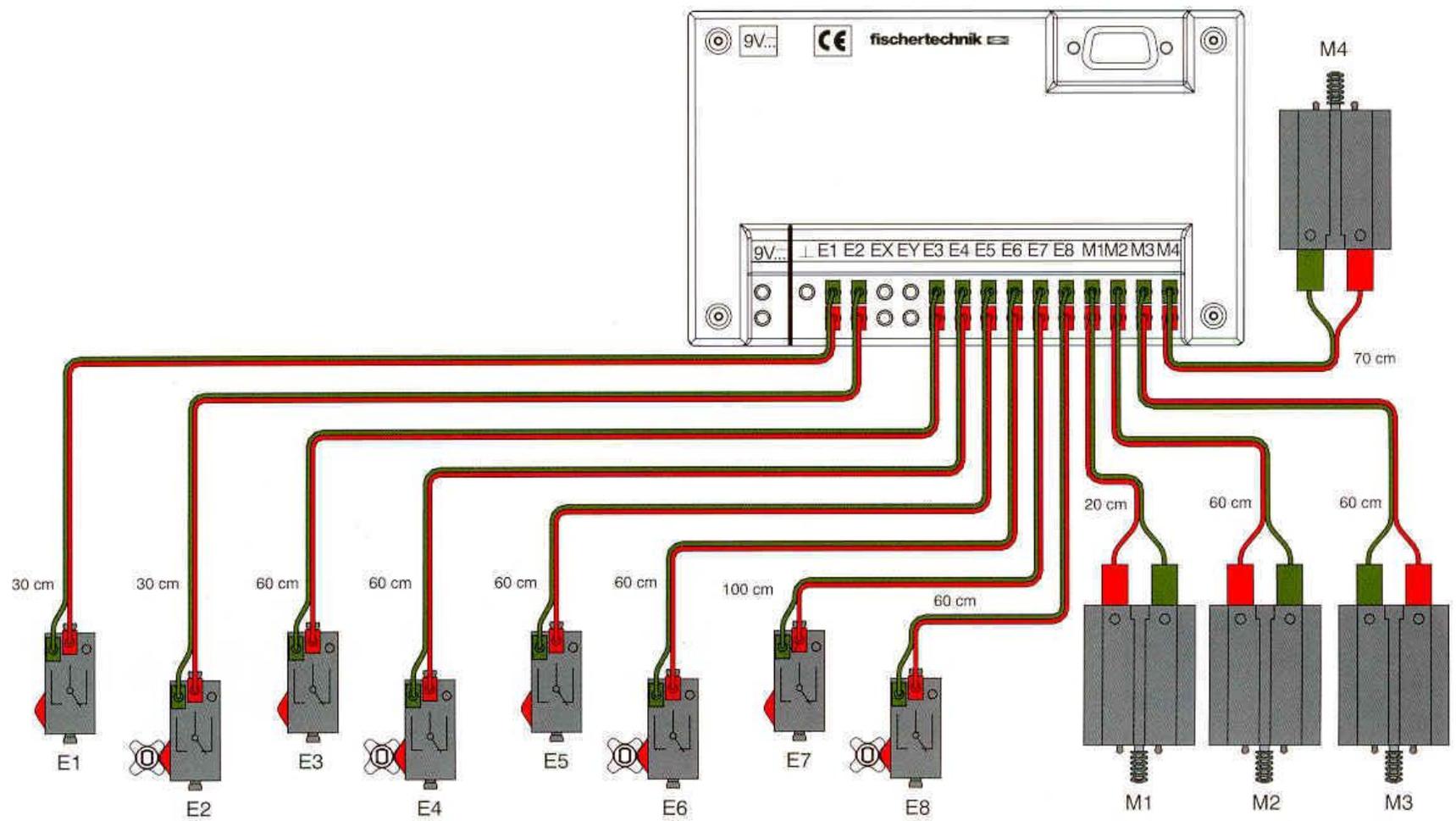
## EL PC COMO SISTEMA DE CONTROL

Puede funcionar en dos modos:

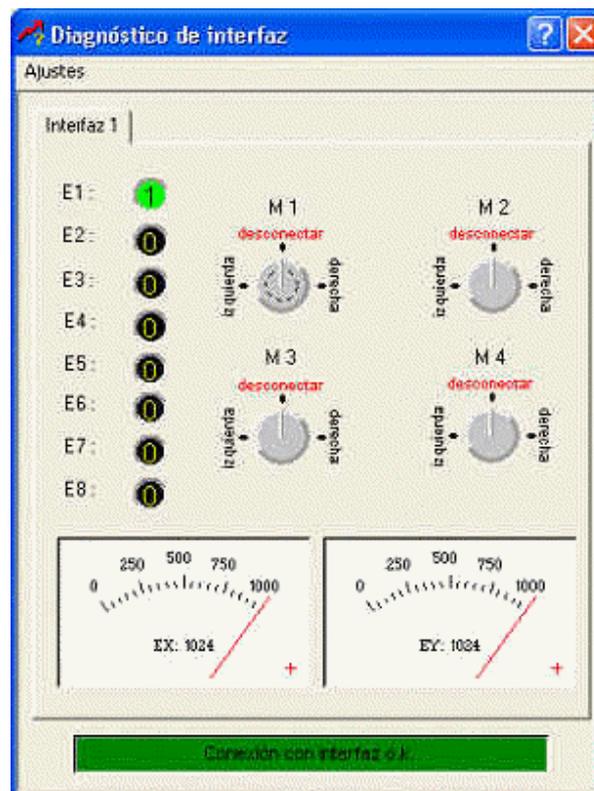
- **modo en-línea:** ("on-line", gobernada en todo momento conectada al PC), o
- **modo descarga:** en este modo el procesamiento del programa tiene lugar únicamente en el microprocesador de la interface, desacoplado del PC. En una primera fase se escribe el programa de control en el PC, luego se descarga en la interface y, finalmente, ésta puede cortar el contacto con el PC (de modo que el cable serie puede ser desconectado).

La forma natural de programar la interfaz es el **lenguaje LLWin de fischertechnik**, pero en modo "on-line" puede ser utilizada mediante cualquier lenguaje de programación. Los detalles acerca de este proceso se pueden encontrar en [www.fischertechnik.de](http://www.fischertechnik.de). En dicha página figuran algunos **drivers** diseñados para diferentes lenguajes de programación. Flowol soporta la programación de esta interfaz sin necesidad de instalar **drivers** adicionales. Sin embargo, **si se utiliza Flowol con esta interfaz, no se dispone del modo descarga.**

Un ejemplo del conexionado de los actuadores y sensores a la interface se muestra en la siguiente figura. En este ejemplo cuatro motores están conectados a las salidas digitales y ocho pulsadores a las entradas digitales. De estos ocho, cuatro se utilizan como finales de carrera (E1, E3, E5, E7) y los otros cuatro (E2, E4, E6, E8) se utilizan como contadores de impulsos (golpeados por una rueda dentada de 4 dientes conforme giran los motores) que sirven para llevar la cuenta de la posición actual del robot - el conjunto motor de c.c. más contador de impulsos actúa como un motor paso a paso-. Este montaje se utiliza en un brazo de robot manipulador articulado de tres grados de libertad finalizado en una pinza.



Las entradas/salidas de la interfaz inteligente se pueden monitorizar/controlar directamente mediante la herramienta **Diagnóstico de interfaz** que se instala conjuntamente con LLWin. Las entradas digitales E1-E8 muestran su estado 0 o 1 (inactivo/activo). Las salidas M1-M4 se pueden desactivar (*desconectar*), polarizar de forma directa (*derecha*, giro horario *-clockwise-* en el caso de un motor) o de forma inversa (*izquierda*, giro antihorario *-counter clockwise-* en el caso de un motor). El valor de las dos entradas analógicas EX y EY se muestra en sendos "vúmetros" con un fondo de escala de 1024 unidades.

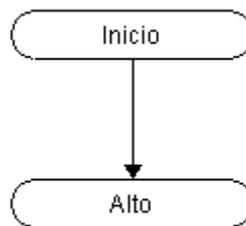


## FUNDAMENTOS DE FLOWOL. EL PROGRAMA

### Introducción

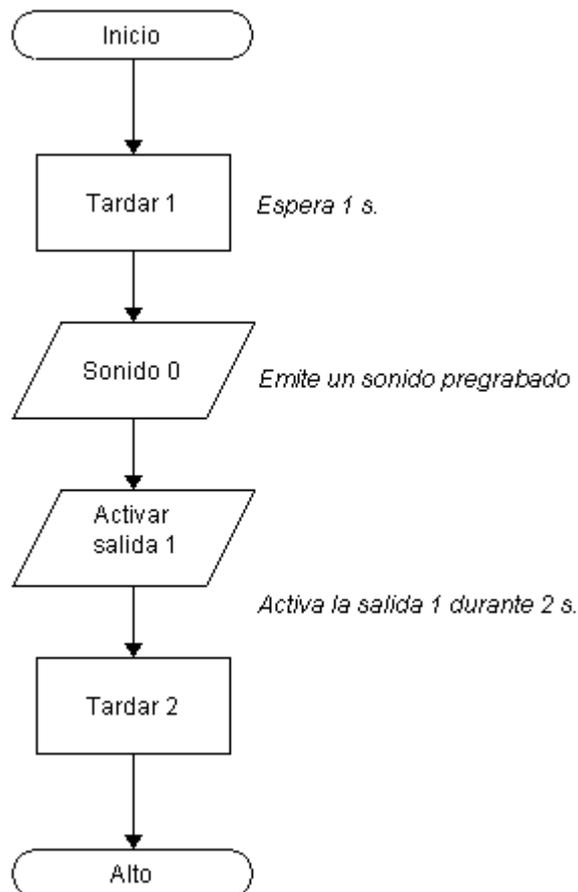
---

La mejor forma de aprender un lenguaje es programando con él. El programa más sencillo que se puede escribir con Flowol es el siguiente:



Como podemos intuir, este programa, con sólo principio (*Inicio*) y fin (*Alto*), no hace nada, pero se ejecuta correctamente y nos da una idea de que no se necesita mucho para empezar a programar en Flowol.

Un programa algo más complicado, pero que hace algo, es el siguiente:



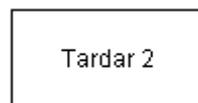
## EL PC COMO SISTEMA DE CONTROL

Con él activamos durante 2 segundos una salida digital de un dispositivo (no especificado por el momento) pasado 1 segundo desde el inicio de la ejecución del programa, y anunciamos la activación de la salida mediante un sonido.

En este programa observamos algunos de los principales elementos de la programación gráfica con Flowol, mediante la sistemática de ordinograma, como son los **bloques terminadores**,



los bloques de realización de un **proceso** (por ejemplo, una espera),



los bloques que producen alguna **salida**,



las **líneas** (flechas) que conectan los distintos bloques y las **etiquetas** (texto en letra cursiva), que constituyen los comentarios del lenguaje.

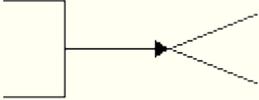
Como se observa, un **programa de Flowol es simplemente una pantalla gráfica en la que se acumulan de forma ordenada bloques de tareas que se conectan secuencialmente mediante líneas** con flechas. No existe un archivo fuente "legible" en el que se encuentre de forma textual el contenido de dicho programa (las instrucciones se encuentran en un formato codificado "no legible").

### Elementos del lenguaje

---

En la siguiente tabla se recogen la nomenclatura y la representación gráfica de los distintos elementos del lenguaje Flowol:

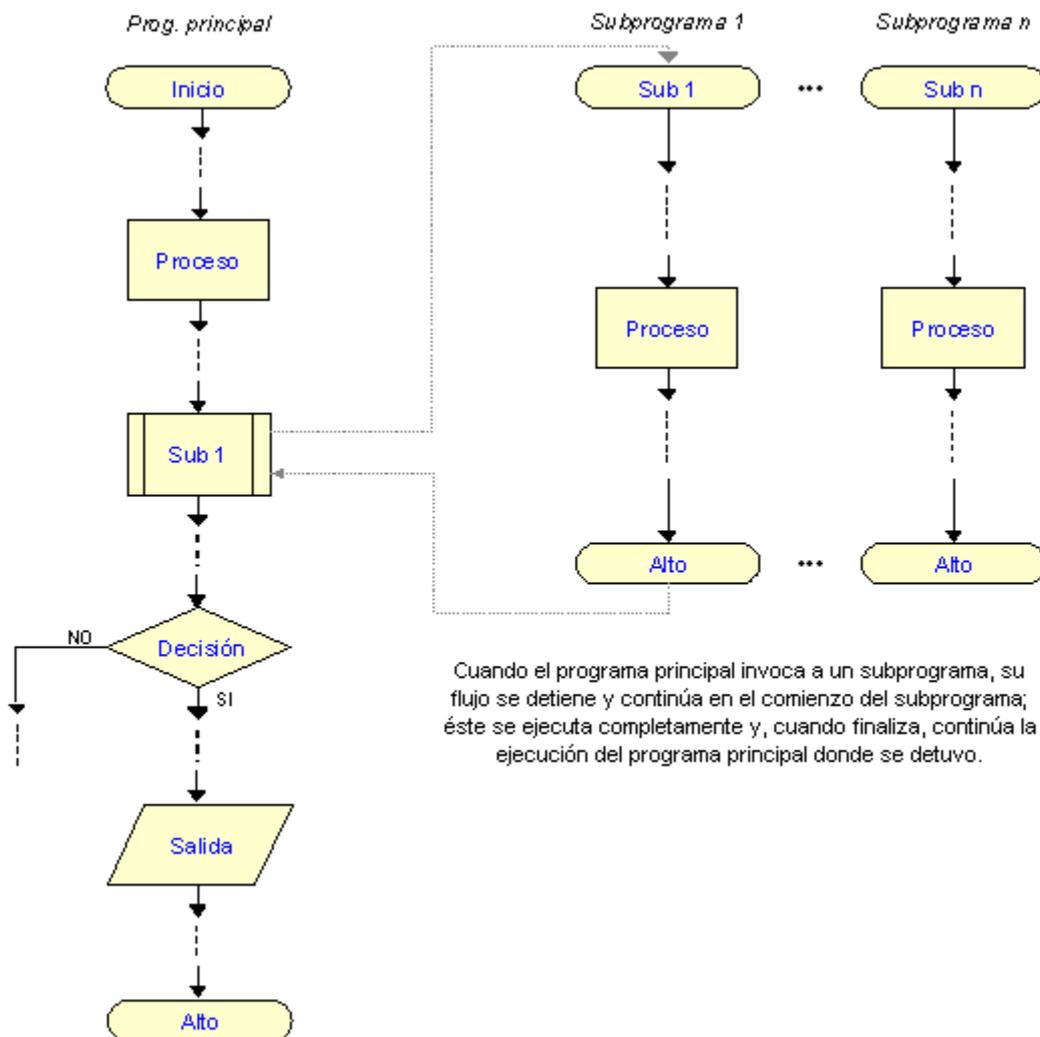
# EL PC COMO SISTEMA DE CONTROL

Elemento	Definición
<p><b>Terminador</b> (<i>Inicio/Alto/Páralo Todo</i>)</p> 	<p>Bloque que <b>marca el inicio o el fin del código de un programa</b> o de un subprograma.</p> <p>Un programa debe empezar necesariamente por un bloque <i>Inicio</i> y terminar con <i>Alto</i>.</p> <p>En caso de que haya más de un bloque Inicio en el programa, de modo que existan varios procesos ejecutándose simultáneamente, el bloque <i>Páralo Todo</i> los detiene a todos.</p>
<p><b>Salida</b></p> 	<p>Bloque que realiza una <b>acción sobre alguna salida de control</b> (activación, desactivación, cambio del sentido de giro de un motor ...).</p>
<p><b>Proceso</b></p> 	<p>Bloque que <b>ejecuta una acción</b> en el programa. Puede ser de tres clases:</p> <ul style="list-style-type: none"> <li>• asignación de valores a variables (bloque <i>Fijar</i>)</li> <li>• retardo (bloque <i>Tardar</i>)</li> <li>• llamada a subprograma (bloque <i>Sub</i>)</li> </ul>
<p><b>Decisión</b></p> 	<p>Bloque que permite <b>bifurcar el flujo del programa en función</b> del resultado de una expresión condicional que depende de:</p> <ul style="list-style-type: none"> <li>• el <b>valor instantáneo de las entradas</b> (analógicas y/o digitales), o</li> <li>• las <b>variables</b></li> </ul> <p>En definitiva, este bloque permite realizar lo que se denomina un <b>control de flujo</b> en el programa.</p>
<p><b>Línea de flujo</b></p> 	<p><b>Conecta secuencialmente dos bloques</b> ejecutables. Puede ser de tres clases:</p> <ul style="list-style-type: none"> <li>• línea ("siempre")</li> <li>• línea "SI"</li> <li>• línea "NO"</li> </ul> <p>Las líneas SI y NO sólo se aplican a las salidas del bloque Decisión, mientras que las líneas ("siempre") se aplican a la entrada y la salida de cualquier bloque (excepto a las salidas del bloque decisión).</p>
<p><b>Etiqueta, comentario</b></p> <p><i>Activa la salida 1</i></p>	<p>No ejecuta ninguna acción en el programa.</p> <p>Se utiliza como un <b>texto explicativo</b> que contribuye a clarificar el código y a recordar más tarde lo que se hizo.</p>

## EL PC COMO SISTEMA DE CONTROL

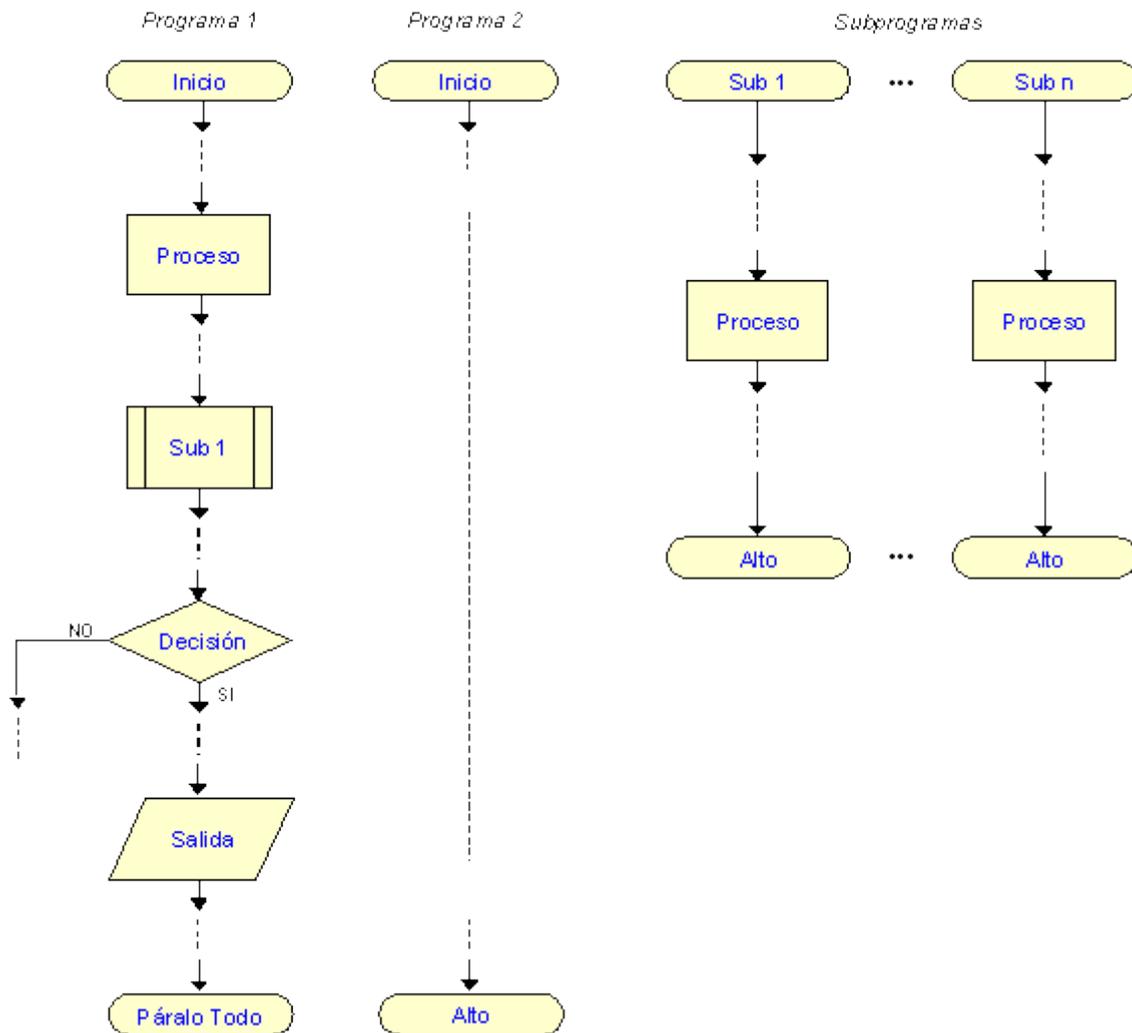
<b><u>Número</u></b>	Designa a valores numéricos enteros que toman valores entre 0 y 9.999.999
<b><u>Variable</u></b>	Un contenedor con nombre para un número. Son el medio para que el programa "recuerde" valores. Se pueden utilizar 8 variables distintas (nombradas como a, b, c, d, n, x, y, z) A la variable se le puede asignar un valor entero, o el resultado de una operación entre dos variables o entre una variable y un número entero.
<b><u>Operador</u></b>	Designa una operación matemática. Los operadores pueden ser <ul style="list-style-type: none"> <li>• aritméticos: +, -, *, /</li> <li>• relacionales: =, !=, &lt;, &gt;</li> </ul>

La estructura de un archivo de programa en Flowol consta esencialmente de un **programa principal** y de varios **subprogramas**:



## EL PC COMO SISTEMA DE CONTROL

Lo anterior no es todavía estrictamente general, dado que aún existe una estructura más amplia, en la que puede darse **paralelismo**, con **varios puntos de inicio simultáneos**; por así decir, **varios procesos o líneas de flujo que se ejecutan simultáneamente** (aparentemente en "en paralelo"):

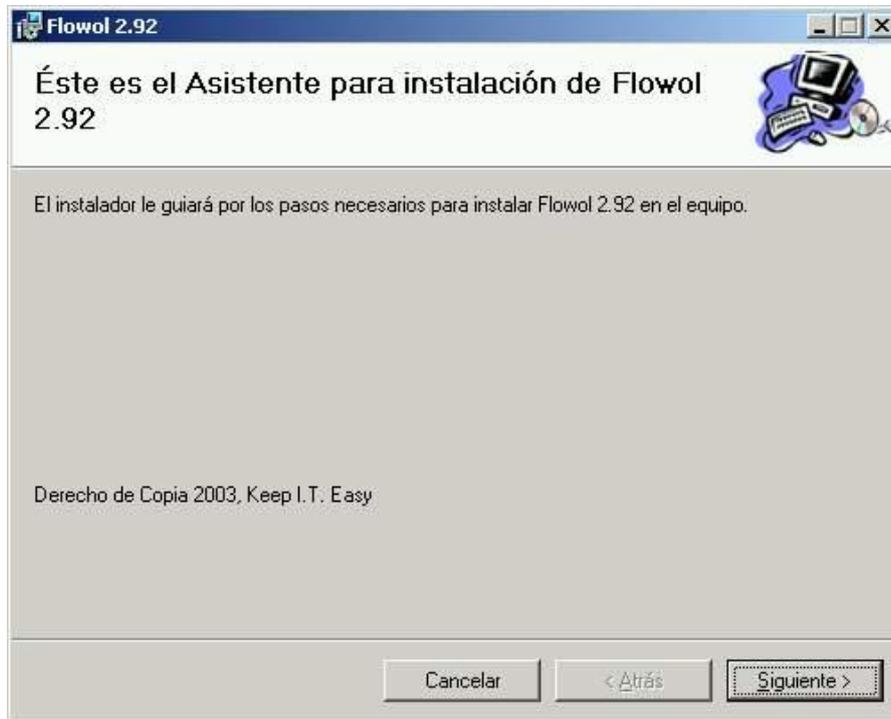


**Nota:** en una estructura como la anterior existen dos o más procesos de programa (4 como máximo) que comienzan a ejecutarse a la vez, y cada uno termina (o continúa indefinidamente) independientemente de los demás. Se puede establecer un mecanismo de sincronismo entre todos ellos utilizando alguna variable "banderín" o "semáforo". El bloque *Páralo Todo* detiene todos los procesos que se hallan en ejecución.

El uso de uno o de varios puntos de inicio en un programa depende del problema concreto de control del que se trate y del tipo particular de solución que quiera o pueda dársele.

## Instalación

El proceso de instalación de Flowol es muy sencillo. Introduciendo en la unidad de CD el disco correspondiente, se inicia automáticamente un asistente de instalación:



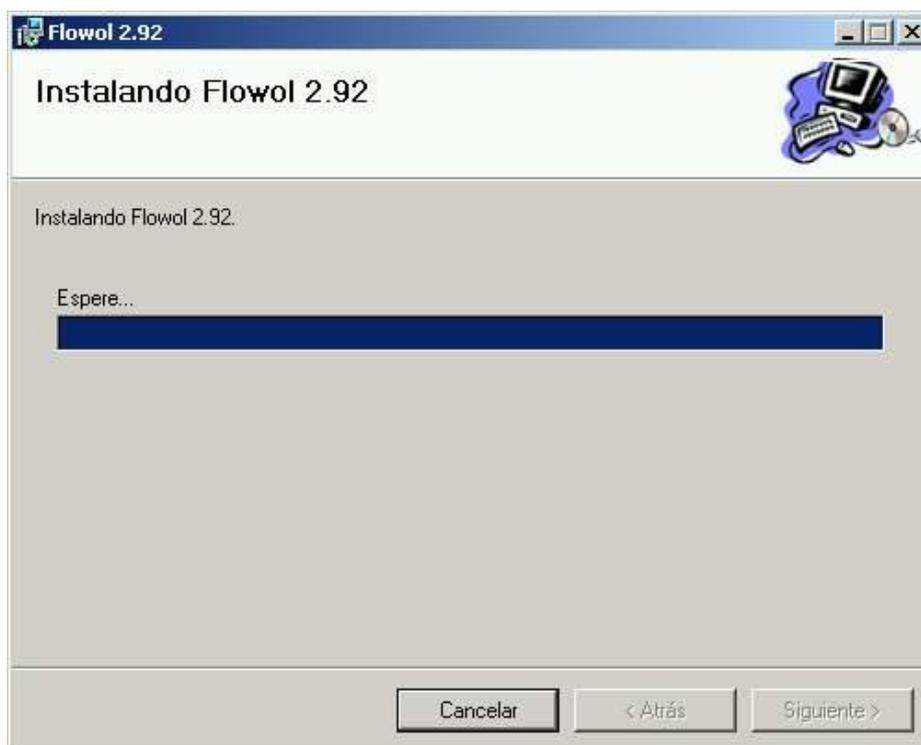
que solicita la carpeta de destino:



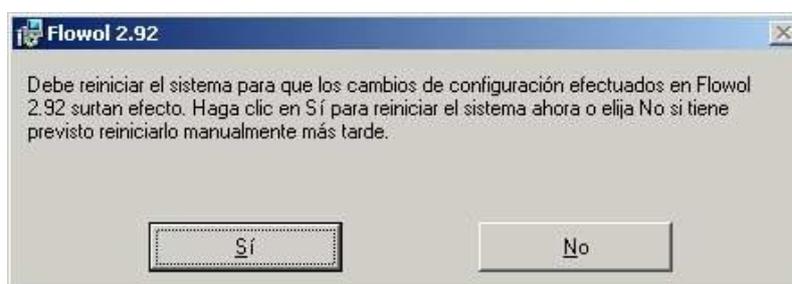
## EL PC COMO SISTEMA DE CONTROL

Un detalle a tener en cuenta: si se instala bajo Windows 2000/XP, se solicita que se especifique si la instalación quedará operativa sólo para el usuario actual o para todos los registrados en el sistema operativo. La opción elegida dependerá del contexto y de la finalidad de la instalación, pero no hay ningún problema en seleccionar "Para todos los usuarios".

Tras pulsar dos veces el botón "siguiente" el proceso de instalación comienza:



y cuando finaliza (tras breves instantes, normalmente unos pocos segundos) se muestra el mensaje:



A partir de los sistemas Windows 2000/XP no es necesario, en circunstancias normales, reiniciar.

Para seleccionar la interfaz de control con la que trabajará Flowol se puede consultar el apartado Selección de la interfaz.

## Edición de un programa

---

Para iniciar un nuevo programa de Flowol se utiliza la opción **Nuevo** del menú *Archivo*. Ésta limpia la ventana de edición si estaba ocupada con un programa anterior:

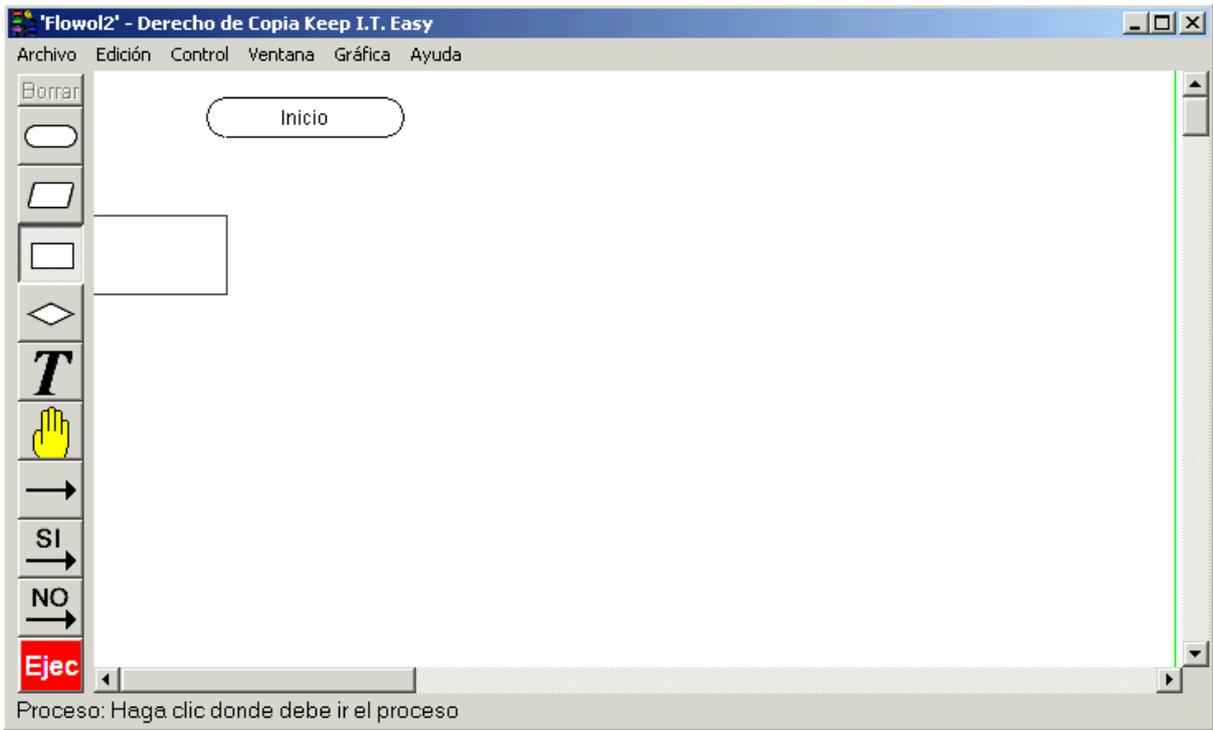


Para introducir un nuevo elemento en un diagrama de programa no hay más que pulsar el botón con el símbolo que identifica el bloque o elemento deseado en la **barra de herramientas**:



Una vez pulsado, al mover el ratón aparecerá un bloque móvil que se puede arrastrar y soltar en el lugar deseado de la **ventana de edición**:

# EL PC COMO SISTEMA DE CONTROL



Una vez colocado en el lugar adecuado, se deben establecer las características particulares del bloque o elemento seleccionado.

Por ejemplo, para insertar un bloque de inicio en un programa, no hay más que pulsar en el símbolo rectangular de esquinas redondeadas de la barra de herramientas, , y, una vez se ha soltado, aparece el siguiente cuadro de diálogo:



Si en éste se pulsa el botón **Inicio**, se obtiene el bloque:



que constituye el punto de arranque de un programa principal. Análogamente, si se pulsa el botón **Alto** se da fin al flujo de un programa.

Si, por ejemplo, el bloque insertado es de *Proceso*, tras soltarlo en el lugar preciso se debe indicar si se trata de establecer un temporizador (*Tardar*), de dar valor a una variable (*Fijar*) o de llamar a un subprograma (*Sub*):

## EL PC COMO SISTEMA DE CONTROL



Y tras ello, aún habría que indicar la duración de la temporización del bloque **Tardar**.

Tras acumular los bloques necesarios en el diagrama para conseguir los efectos deseados, hay que **unirlos con las líneas** pertinentes, que deciden por dónde transcurrirá el flujo o la secuencia del programa. Para introducir alguna de dichas líneas, no hay más que hacer clic en el botón correspondiente de la barra de herramientas:



y tras ello, pulsar en el bloque origen y en el bloque destino consecutivamente. Las **líneas se dibujan automáticamente** siguiendo un recorrido determinado por Flowol. Si se desea **dibujar las líneas siguiendo un recorrido elegido por el usuario** no hay más que pulsar primeramente en el bloque origen y, seguidamente, ir haciendo clic en los distintos puntos del área de edición por donde se desea que transcurran las líneas de flujo; finalmente, hay que hacer clic en el bloque destino.

Los **comentarios del programa** se introducen pulsando el botón  de la barra de herramientas y escribiendo texto en la forma habitual. Cada comentario puede tener un máximo de 56 caracteres.

Las **características de un bloque insertado previamente se pueden cambiar en cualquier momento**. Para ello hay que **seleccionarlo**; ello se consigue pulsando el icono *Mano*:



y haciendo clic con el ratón sobre el bloque deseado (o también, haciendo directamente clic sobre el bloque deseado con el botón derecho del ratón). Se abrirá entonces un cuadro de diálogo que permitirá cambiar las propiedades del bloque. Este cuadro

## EL PC COMO SISTEMA DE CONTROL

depende de cada bloque en particular y se trata en los distintos apartados de esta documentación.

**Nota:** para modificar un bloque es preciso borrar primero sus propiedades anteriores para volver a definir las. Esto se puede hacer con el botón *Borrar* del cuadro de diálogo de Edición.

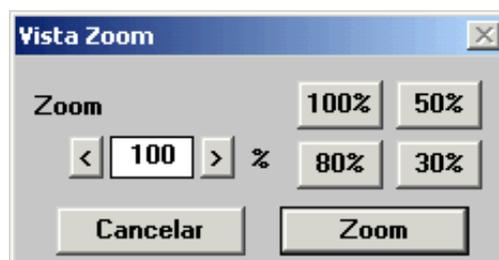


Si lo que se desea es **eliminar completamente un bloque o una línea** del diagrama de programa, se debe seleccionar primero el elemento deseado con el icono *Mano* y borrarlo después pulsando el botón *Borrar*.



Igualmente, se pueden **borrar un conjunto de elementos** seleccionándolos simultáneamente (pulsar primero el icono *Mano* y después pulsar sobre la ventana de edición arrastrando el ratón hasta que queden seleccionados todos los elementos deseados -aparece un cuadro que los engloba-) y haciendo clic sobre el botón *Borrar*.

Cuando el número de símbolos en un programa empieza a ser importante, éste puede superar los límites de la ventana de edición, de modo que no pueda ser observado al completo. En tal caso, es útil usar la opción *Vista Zoom...* del menú *Ventana*:



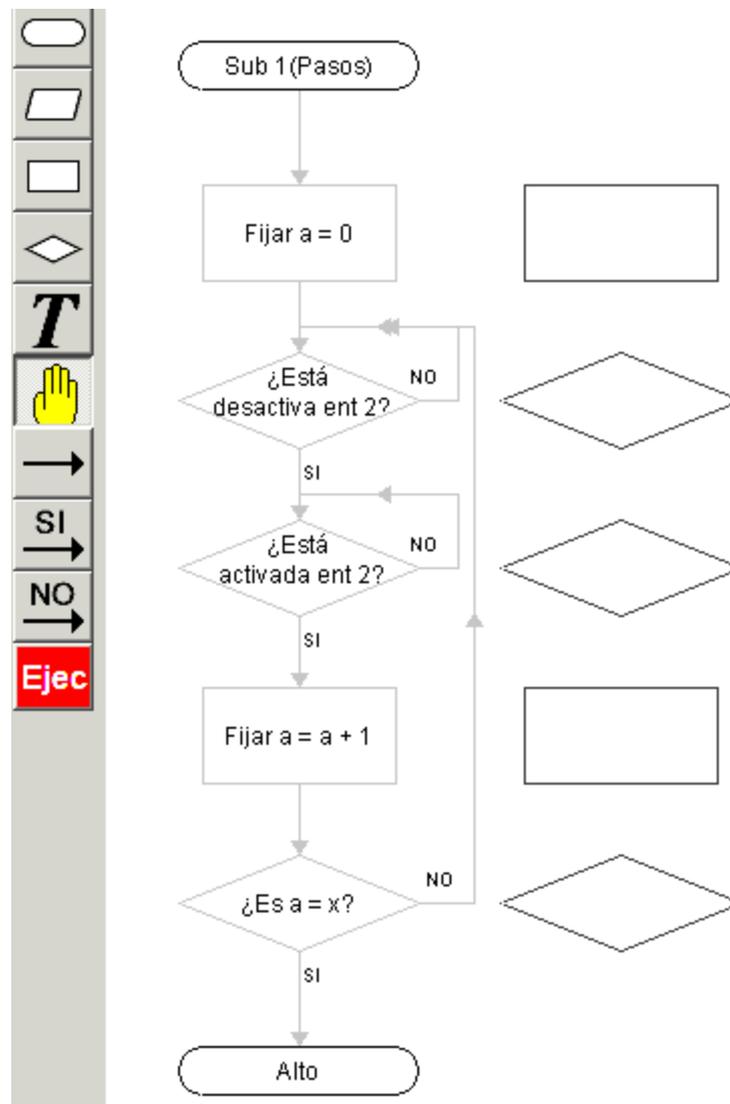
que permite ampliar/reducir el área visible de edición, según las necesidades.

Cuando se ha terminado la edición del programa, éste se puede **almacenar en disco** con las opciones **Guardar y/o Guardar como...** del menú **Archivo**. Una vez elegido un nombre representativo para el programa, éste se guarda en la carpeta de disco deseada con extensión **.flo**. Cuando se desee acceder de nuevo a este programa, se puede abrir desde

## EL PC COMO SISTEMA DE CONTROL

dentro de la interfaz gráfica de Flowol con la opción **Abrir** del menú **Archivo**, o haciendo directamente doble clic sobre el archivo en el disco (sin abrir antes Flowol), de modo que se inicia Flowol con el programa cargado en la ventana de edición.

Cuando se editan programas con cierto grado de complejidad, resulta de ayuda la utilización de la función de **copiado y pegado de un conjunto de bloques**. Supongamos que se quiere duplicar un conjunto de bloques para reutilizarlos en una sección distinta del programa. Para conseguirlo no hay más que seleccionarlos mediante el icono *Mano*, seguidamente arrastrarlos mientras se mantiene pulsada la tecla *Shift* ( *Y* ), y soltarlos en el punto deseado de la ventana de edición; durante el arrastre los bloques originales seleccionados se marcan en gris y los bloques duplicados se marcan mediante su contorno, como se observa en la siguiente figura:



**Nota:** esta función no permite copiar un bloque o conjunto de bloques y pegarlo desde un programa a otro diferente. El copiado y pegado se realizan sólo dentro de un mismo programa.

## Ejecución de un programa

---

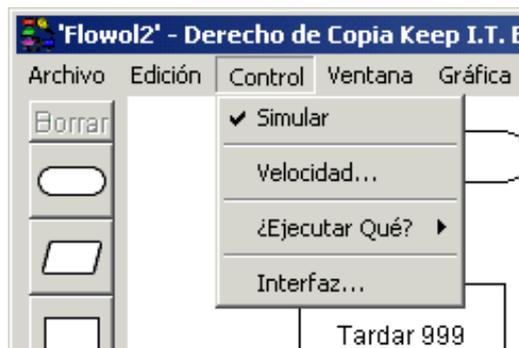
Para llevar a cabo las acciones codificadas en un programa (ejecución del programa), no hay más que tenerlo cargado en la ventana de edición y pulsar el botón "Ejecución" de la barra de herramientas:



La ejecución se puede producir en dos modos:

- modo simulación
- modo "en-línea" o activo

En el **modo simulación** no se envía ninguna orden a la interfaz de control, de modo que se puede comprobar el funcionamiento del programa sin ningún efecto físico. Para entrar en modo simulación (el modo por defecto cuando se inicia Flowol), no hay más que marcar la opción *Simular* del menú *Control*.



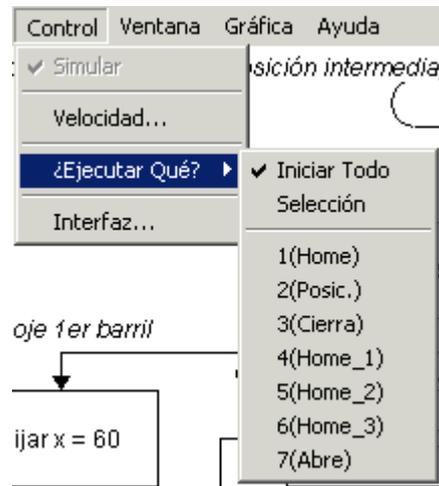
Existe una extensión de Flowol para el modo simulación denominada **Mimics**. Se trata de un módulo de *software* que permite simular la ejecución de un programa sobre un "fondo o maqueta virtual", comprobando los efectos del programa sin necesidad de dispositivos físicos. Los detalles de este módulo se pueden ver en la página 38 (Simulación con Mimics).

Para ejecutar el programa **enviando las órdenes a la interfaz de control (modo "en-línea" o activo)**, no hay más que quitar la marca de la opción **Simular** del menú **Control**.

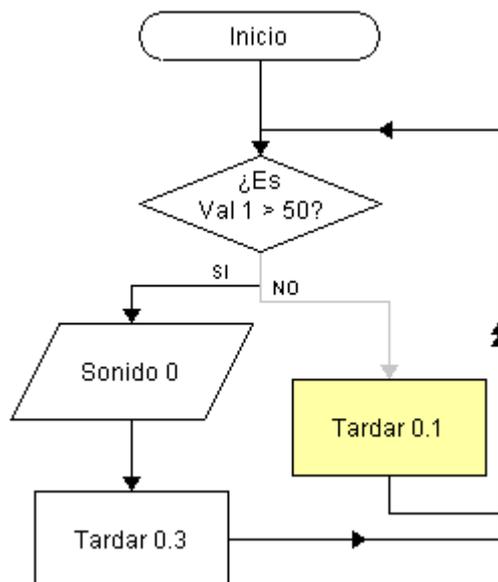
Si se desea, no hay necesidad de ejecutar el programa como un todo; **se pueden ejecutar independientemente un conjunto de bloques seleccionado o algún subprograma en**

## EL PC COMO SISTEMA DE CONTROL

particular. Para ello, no hay más que acudir a la opción *¿Ejecutar Qué?* del menú **Control**. El menú desplegable que aparece permite elegir entre la totalidad del programa, los bloques seleccionados y los subprogramas:



Durante la ejecución de un programa o de una sección del programa se puede observar el flujo del mismo a tiempo real puesto que se colorea de forma especial el bloque que se halla en ejecución en cada instante, así como la línea de flujo correspondiente.



En el programa de la figura anterior observamos que en el instante considerado el flujo viene de la salida *NO* del bloque Decisión y que actualmente se está ejecutando un bloque de proceso **Tardar 0.1** (con una temporización de 0.1 segundos).

## EL PC COMO SISTEMA DE CONTROL

Como se acaba de decir, Flowol colorea las líneas y los bloques durante la ejecución; pero si en la ventana **Control de velocidad** (ir a menú **Control**, opción **Velocidad...**) se quitan las marcas de las casillas **Líneas** y **Símbolos** (que Flowol pone por defecto):



entonces ya no se colorean líneas ni bloques; esto aumenta la velocidad de ejecución del programa. La velocidad se puede regular también con la barra deslizante de la parte superior, que por defecto está anclada en el 80% de la máxima posible, cuando se inicia Flowol.

Es posible lanzar un programa en modo **paso a paso**, de manera que se ejecute un solo bloque cada vez. Ello se consigue marcando la casilla *Barra Espaciadora* de la ventana anterior, de modo que tras ejecutar un bloque no se pasa al siguiente hasta que se pulsa la barra espaciadora del teclado.

Si se desea **detener un programa en ejecución antes de que termine su secuencia lógica** (es decir, antes de alcanzar un terminador *Alto* o *Páralo todo*) se puede pulsar el botón *Alto*:



de la barra de herramientas.

Con algunas interfaces de control, todavía se puede ejecutar un programa de un tercer modo:

- **modo descarga** (descargándolo previamente a la memoria de la interfaz).

Esto es cierto, en particular, con la interfaz FlowGo. Para ello, se debe pulsar el botón que aparece a tal efecto bajo el botón **Ejecutar** con el nombre de la interfaz:



## EL PC COMO SISTEMA DE CONTROL

Esto desencadenará un proceso de compilación del diagrama de flujo y de descarga en la memoria de la interfaz:



Cuando el proceso de descarga finaliza:



el programa se puede ejecutar desde la interfaz, incluso desconectado del ordenador, sin más que pulsar en el botón que a tal efecto se encuentra en la interfaz de control (en el caso de FlowGo existe un botón denominado *Go*).

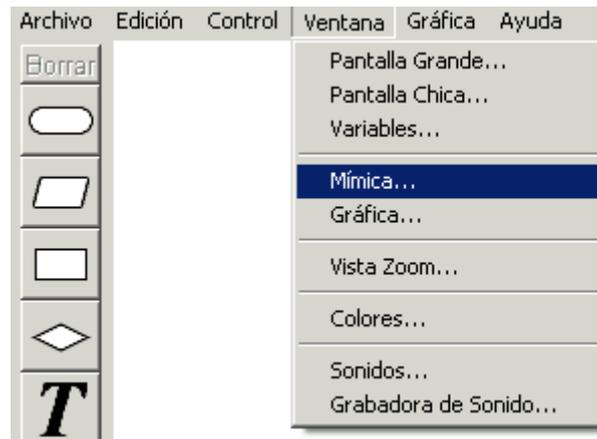
### Simulación con Mimics

---

Flowol cuenta con un módulo denominado **Mimics** que permite realizar simulación utilizando una especie de "fondos o maquetas virtuales" que ofrecen una imagen representativa de un sistema real y la posibilidad de manipular mediante *software* entradas y salidas ubicadas sobre la imagen.

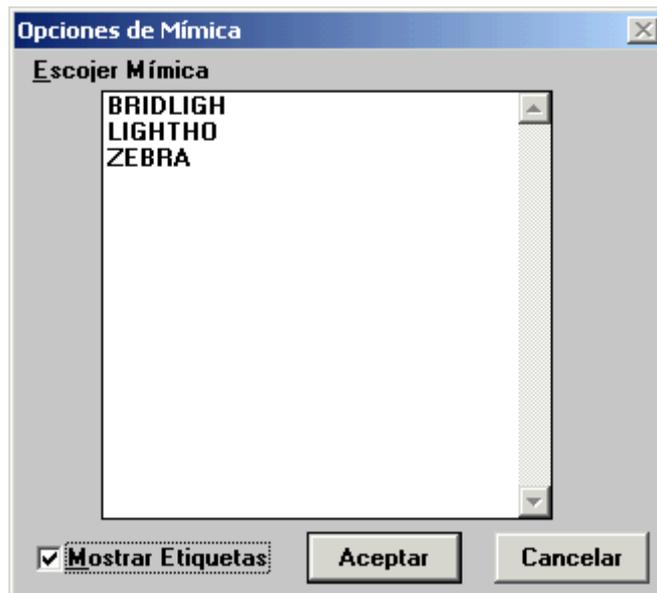
Para cargar una mímica se utiliza el menú **Ventana/Mímica**:

## EL PC COMO SISTEMA DE CONTROL



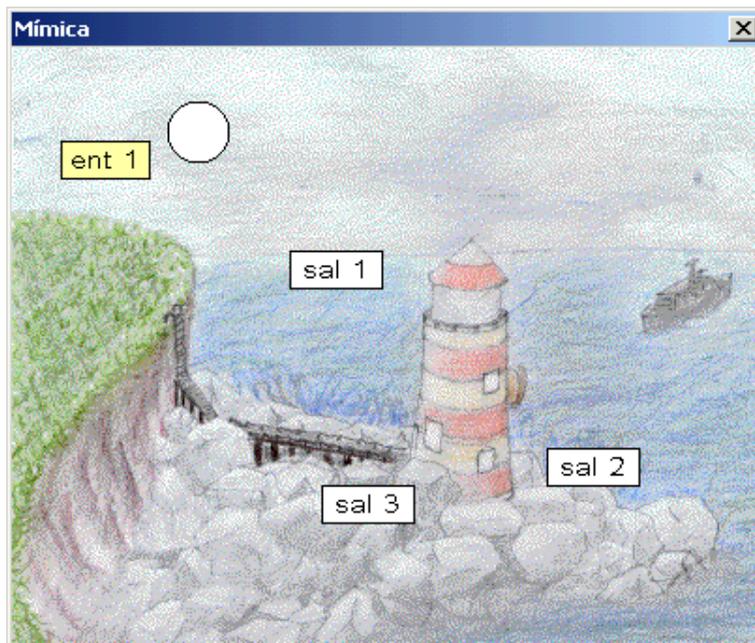
Por defecto, Flowol incluye tres archivos de Mímics:

- **Bridligh:** un cruce de dos semáforos (6 salidas = 3 luces de ambos semáforos)
- **Lightho:** un faro (tres salidas: luz de faro, luz en las escaleras de la torre y aviso acústico; 1 entrada: detección de día/noche -sol iluminando/no iluminando).
- **Zebra:** una señalización de semáforo con una sola luz ámbar (una salida)



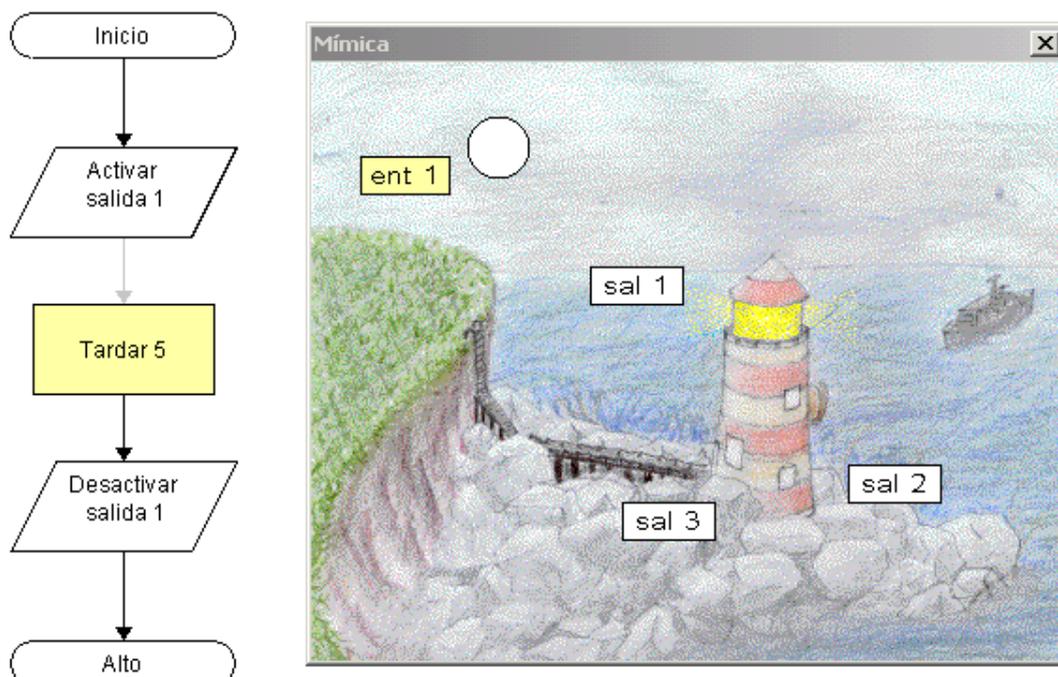
Es conveniente marcar la casilla **Mostrar Etiquetas** para que sobre la imagen del archivo de mímica se indique dónde se encuentran las entradas/salidas. Por ejemplo, si se selecciona *LIGHTHO* y se pulsa *Aceptar* se carga la ventana:

## EL PC COMO SISTEMA DE CONTROL



donde se observa que al sol se le asigna una entrada digital (*ent 1*) que indica si existe luz ambiente o no día/noche). Asimismo, se asignan tres salidas digitales: a la luz del faro (*sal 1*), a la bocina del faro (*sal 2*) y a la iluminación de las escaleras de la torre (*sal 3*).

Un programa muy sencillo de ejemplo de uso de esta mímica puede ser el siguiente:



Se observa que en éste sólo se activa la luz del faro (obsérvese luz amarillenta adjunta a *sal 1*). Lógicamente esta mímica está pensada para programar un proceso de realimentación en el que se detecte si es de día o de noche y en función de ello

## EL PC COMO SISTEMA DE CONTROL

encender/apagar la luz del faro y de la torre. Asimismo, se puede "jugar" con la presencia o ausencia de barco para activar la bocina del faro.

Se pueden adquirir mediante compra otros archivos de mímica pensados para Primaria y Secundaria. En el manual de Flowol (zip, 1.7 MB) se pueden encontrar detalles al respecto.

Asimismo, existen mímicas para simular los diferentes montajes que se pueden realizar mediante el *Fischertechnik Starter Pack*, que se encuentran disponibles de forma gratuita con la adquisición del Flowol.

Los archivos de mímica pueden ser creados de forma relativamente sencilla por uno mismo. En el documento *Cómo crear Mímicas con Flowol 2.91* -un estupendo trabajo final del presente curso de Control y Robótica-, Fernando Ledesma (IES Zorrilla, Valladolid) describe cómo hacerlo y lo aplica a una mímica de su creación denominada ascensor.