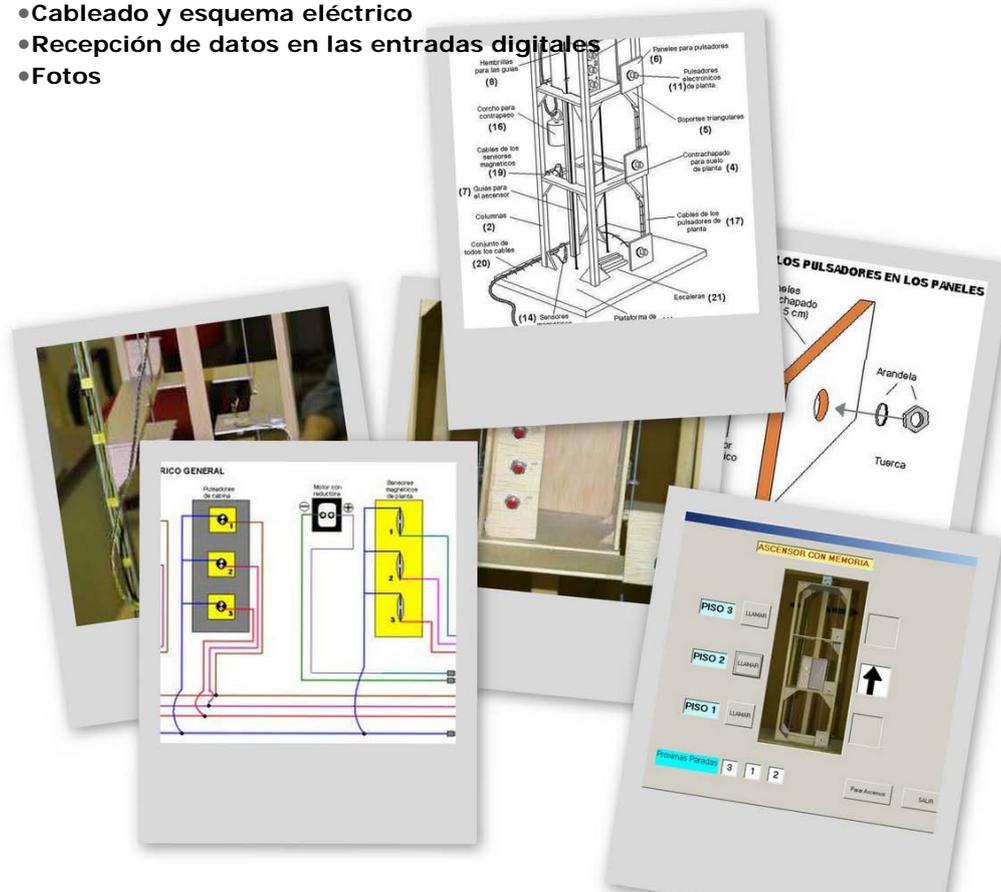


ASCENSOR

- Introducción
- Descripción de elementos
- Cableado y esquema eléctrico
- Recepción de datos en las entradas digitales
- Fotos



INTRODUCCION

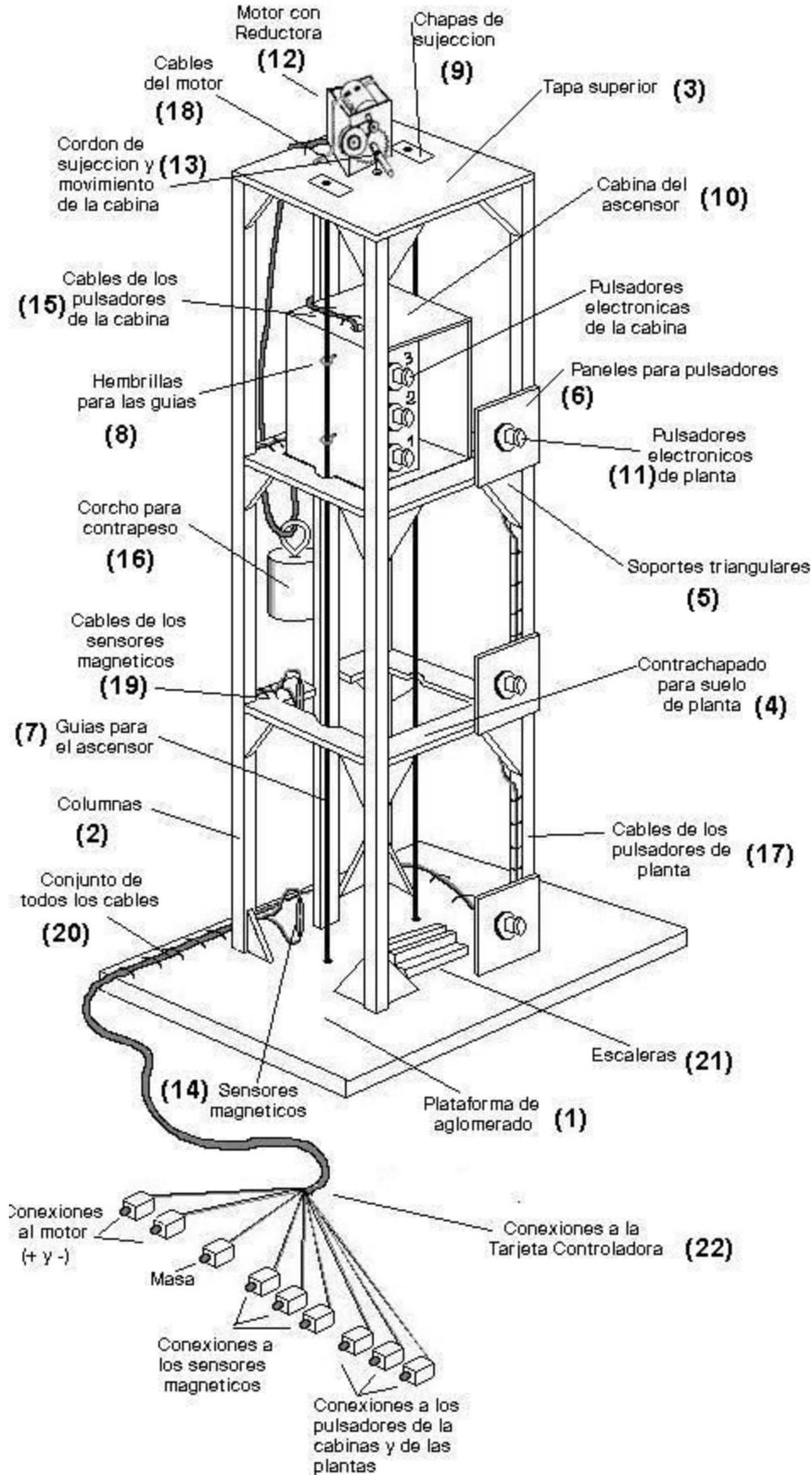
Este dispositivo simula el funcionamiento de un ascensor de 3 plantas. Como en la realidad, el ascensor podrá ser llamado desde cada una de las plantas por medio de unos pulsadores eléctricos; del mismo modo, desde dentro de la cabina podremos elegir el piso de destino. También, el ascensor almacenará las distintas llamadas que se puedan producir desde cada planta, e ira moviéndose en función del orden de llamada.

El sistema se ha construido con un único motor, que realiza el movimiento de subida y bajada de la cabina del ascensor. En cada planta, hay colocado un sensor magnético, que detecta cuando pasa la cabina por el, para pararlo en caso de que se haya realizado la llamada, en la misma planta que la del sensor.

DESCRIPCION DE ELEMENTOS

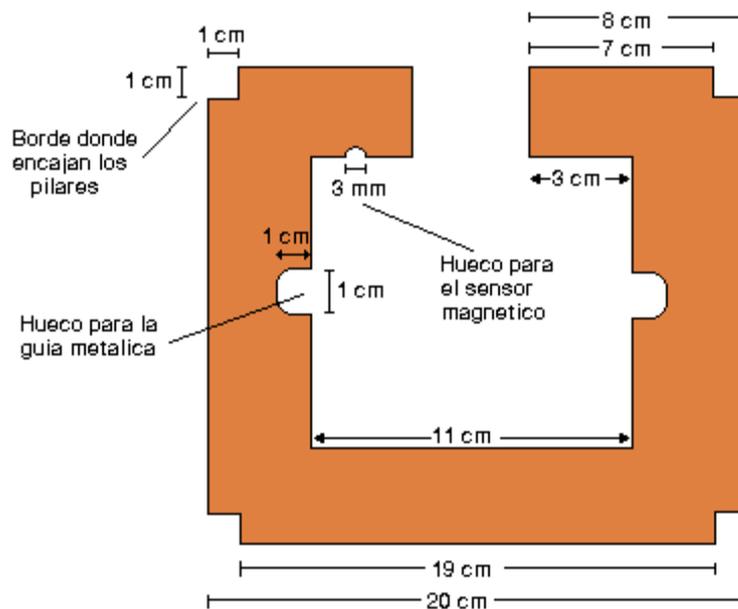
Toda la estructura, se apoya en una **Plataforma de Aglomerado (1)** de 30 x 40 x 1 cm. Los elementos que la componen son:

DESCRIPCION DE ELEMENTOS DEL ASCENSOR

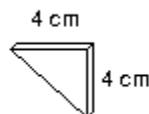


- **Columnas (2):** Los pilares de la estructura son 4 maderas de 20 cm de alto, 1 cm de ancho y 1 cm de largo. están situadas en las esquinas de un cuadrado imaginario, a 20 cm de separación entre cada una de ellas.
- **Tapa superior (3):** Se apoya encima de los 4 pilares. Se trata de una plancha de contrachapado de 20 x 20 cm, y en el se apoya el motor y cuelgan las guías de la cabina.
- **Contrachapados para suelo de planta (4):** Contrachapado de 20 x 20 cm, que sirve de suelo para las plantas primera y segunda. Posee un agujero rectangular de 11 cm de lado, por el que pasa sobradamente la cabina cuando se desplaza entre los pisos. La estructura de esta pieza se detalla en la imagen.

CONTRACHAPADO PARA SUELO DE PLANTA



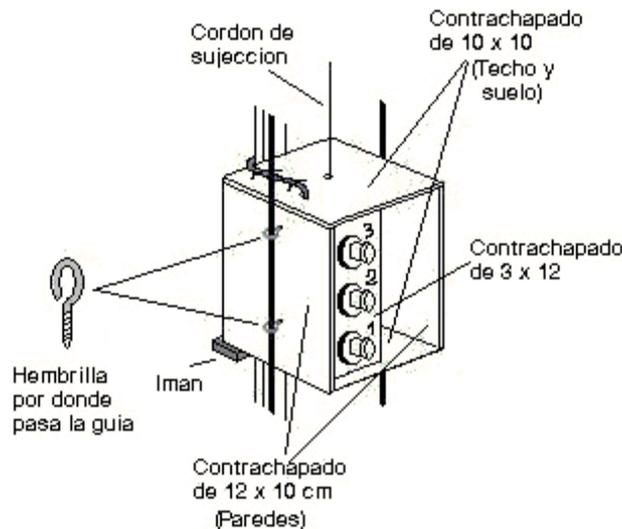
- **Soportes triangulares (5):** Triangulo rectángulo cuyos catetos miden 4 cm y sirven para dar soporte al suelo de cada piso. están pegadas a las columnas de la estructura, y hay 8 por cada piso.



- **Paneles para pulsadores (6):** Contrachapado rectangular de 4 cm de lado. Sirve de sujeción para los pulsadores electrónicos. Hay un panel por cada planta, y va fijado entre el lado de una de las columnas, y el suelo.
- **Guías para el ascensor (7):** Hay de dos tipos:
 - Una guía de madera de 60 cm de longitud, con una sección rectangular de 1 cm de ancho y 2 cm de largo. Está situada en la parte trasera de la estructura, y sujeta a la cabina en su parte posterior por medio de un enganche de cartón fijado a ésta.
 - 2 secciones de alambre de 2 mm de espesor y 61 cm de longitud, puestos verticalmente dentro de la estructura, y situados a ambos lados de la cabina. Están introducidas medio centímetro en la plataforma y sujetas en el techo por medio de unas chapas de sujeción metálicas (9) de 2 x 1.5 cm. Cada alambre, está fijado a la cabina por medio de 2 hembrillas (8), que son anillos de metales, por donde pasa la guía, y que van clavados a la cabina. Las hembrillas tienen una separación entre sí de 5 cm.

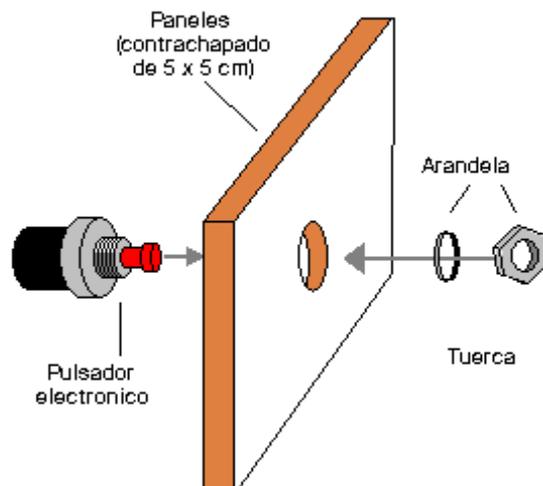
Interfaz de control de dispositivos externos por ordenador a través de puerto paralelo

- **Cabina del ascensor (10)** : octaedro de 12 cm de alto, 10 cm de ancho y otros 10 cm de profundidad. Uno de sus lados está abierto a modo de puerta, y posee 3 pulsadores electrónicos para realizar las llamadas a pisos desde la cabina. En la cara inferior del suelo, en la parte trasera, se ha instalado un pequeño imán para que sea detectada la cabina por los sensores magnéticos situados en las distintas plantas (14).



- **Pulsadores electrónicos (11)**: Dispositivos que mandan un impulso eléctrico de 5 V al ser presionados. Hay tres pulsadores que están encajados en los paneles de cada planta, y otros tres situados en la cabina. Cada uno de ellos, mueve la cabina a la planta correspondiente. Hay que tener en cuenta, que por ejemplo, el pulsador de la planta 1, tiene el mismo efecto que el pulsador 1 de la cabina.

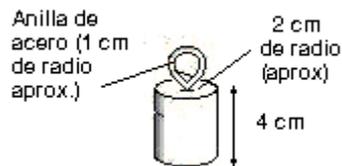
MONTAJE DE LOS PULSADORES EN LOS PANELES



- **Motor con reductora (12)**: Situado encima de la tapa superior. En uno de sus ejes está enrollado el **cordón de sujeción de la cabina (13)**, que según en qué sentido gire el motor, recogerá o soltará cuerda, de manera que la cabina, subirá o bajará. Este cordón ha de tener una longitud de algo mas de 40 cm, y se introduce por un agujero hecho en el centro de la tapa superior, quedando unido a la cabina también en el centro de su tapa superior.
- **Sensores magnéticos (14)**: Dispositivos que emiten una señal eléctrica de 5 V cuando se le somete a una campo magnético. Hay tres, y están situadas entre medias del contrachapado que representa al suelo de cada planta. Cuando la cabina, en su movimiento vertical, pasa por una planta, activa el sensor magnético mediante el imán

instalado en su suelo. De este modo, se crea una señal que ha de ser interpretada posteriormente, con el fin de decidir si la cabina se para o, por el contrario, prosigue su camino.

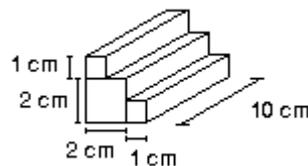
- Con respecto a los cables deberemos hablar de cuatro agrupaciones:
 - Los **cables de los pulsadores de la cabina (15)** van enganchados a una de las guías del ascensor, y cuelgan por medio de un **corcho (16)**. Estos cables suben hasta el techo de la estructura. Su longitud debe ser la suficiente como para que si la cabina está en el primer piso, los cables lleguen arriba. De esta forma, se permite el movimiento de la cabina y el de sus cables. Estos cables se fijan en el techo del ascensor y descienden por una de los pilares.



- Los **cables de los pulsadores de planta (17)** descienden por el pilar más cercano hasta el suelo.
- Los **cables del motor (18)**, descienden por uno de los pilares, el mismo por el que descienden los cables de los pulsadores de la cabina.
- Los **cables de los sensores magnéticos (19)** descienden por otro de los pilares traseros hasta la plataforma.

Todos ellos, se agrupan en un **conjunto de cables (20)**, situados en la plataforma.

- **Escaleras (21):** Objeto decorativo, situado en la planta baja, que queda justo delante de la cabina cuando se detiene en dicha planta. Está formada por 2 piezas de madera de 10 cm de longitud por 1 cm de ancho y fondo (que hacen de escalones 1 y 3), y otra pieza, igualmente de madera, de longitud 10 cm, y 2 cm de ancho y fondo (que hace de escalón 2 y apoyo del escalón 3).



- **Conexiones a la tarjeta controladora (22):** Los distintos conectores que lleva el ascensor son:
 - 2 conectores del motor, positivo y negativo.
 - 1 conector de masa.
 - 3 conectores de los sensores magnéticos, uno por cada piso.
 - 3 conectores de los pulsadores de las cabinas y los pisos, teniendo en cuenta, que por ejemplo, el pulsador del piso 1, y el botón 1 de la cabina, producen el mismo efecto, y por ello, se unen en un solo conector.

CABLEADO Y ESQUEMA ELECTRICICO

La imagen de la derecha, siguiente la vista trasera del ascensor, e indica la forma en como los cables se sitúan en la estructura.

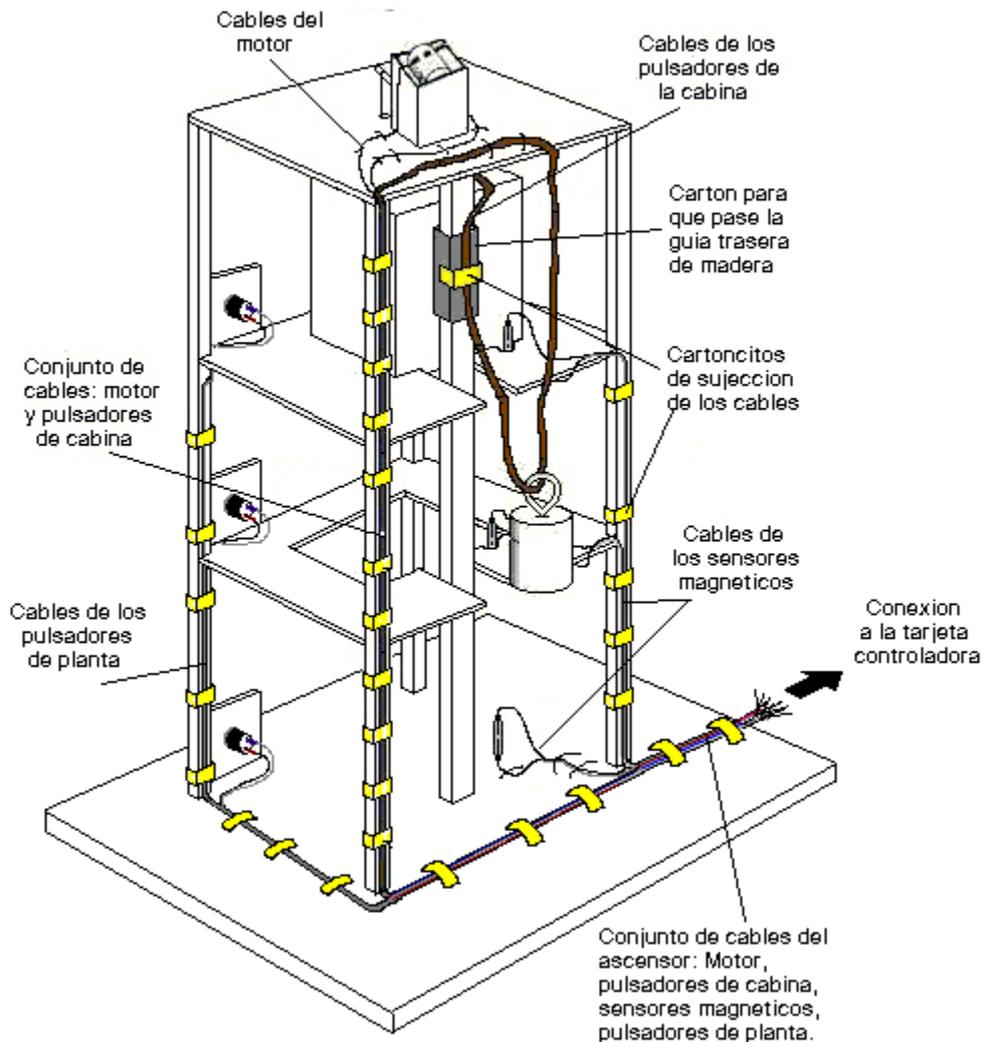
Los cables están fijados mediante pequeños cartoncitos, pegados a las maderas, o mediante grapas.

Interfaz de control de dispositivos externos por ordenador a través de puerto paralelo

La longitud aproximada de cable, para cada grupo, es la siguiente:

- Cables del motor: 2 cables (positivo y negativo) de 120 cm cada uno.
- Cables de los pulsadores de planta: Para el pulsador 1, 70 cm, para el pulsador 2, 90 cm, y para el pulsador 3, 110 cm.
- Cables de los pulsadores de la cabina: Más o menos, 180 cm para los 3 pulsadores.
- Cables de los sensores magnéticos: Para el sensor 1, 30 cm. Para el sensor 2, 50 cm, y para el sensor 3, 70 cm.

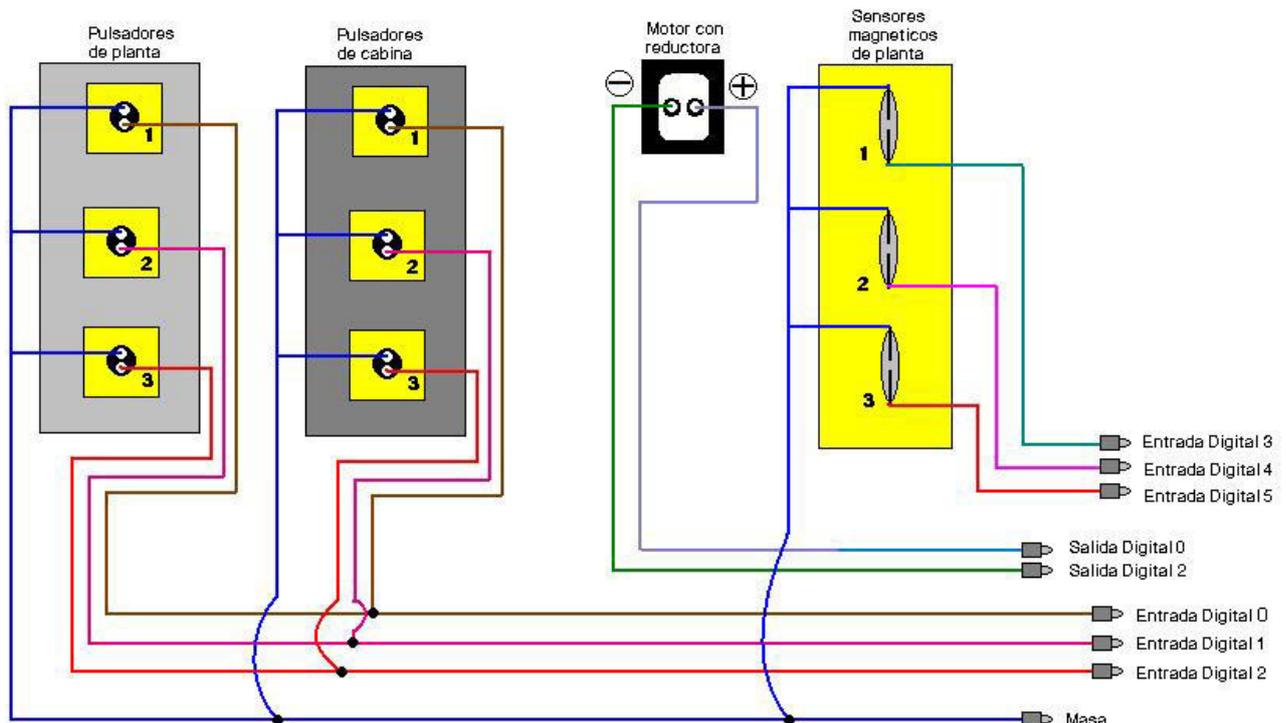
COLOCACION DE LOS CABLES POR LA ESTRUCTURA - VISTA TRASERA



En cuanto al esquema eléctrico (reflejado en el esquema inferior) debemos destacar la unión de los cables de masa de cada componente en una única conexión, ya que no es necesario que haya mas.

Los pulsadores de planta y de cabina, que llaman al mismo piso, tienen efectos iguales, por lo que también se han unido derivando en un único cable. Así, que solo habrá 3 cables que representan a los pulsadores, tanto de planta, como de cabina.

CIRCUITO ELECTRICO GENERAL



RECEPCION DE DATOS EN LAS ENTRADAS DIGITALES

Teniendo en cuenta como se han realizado las conexiones entre la tarjeta controladora y el ascensor, los datos recibidos en las entradas digitales, cuando se produce un evento, son los siguientes:

Acción	Dato que se recibe en las entradas digitales
Presión del Botón 1 (Cabina y planta)	1
Presión del Botón 2 (Cabina y planta)	2
Presión del Botón 3 (Cabina y planta)	4
Activación del Sensor Magnético de la planta 1	8
Activación del Sensor Magnético de la planta 2	16
Activación del Sensor Magnético de la planta 3	32
La cabina está entre plantas	0

Ahora bien, puede haber distintas combinaciones que difieran de los valores anteriores, por ejemplo, en el instante en que se pulsa uno de los botones y a la vez se active uno de los sensores magnéticos de una planta. La relación de los datos recibidos en las entradas digitales, y las acciones a realizar con la cabina, viene reflejado en la siguiente tabla.

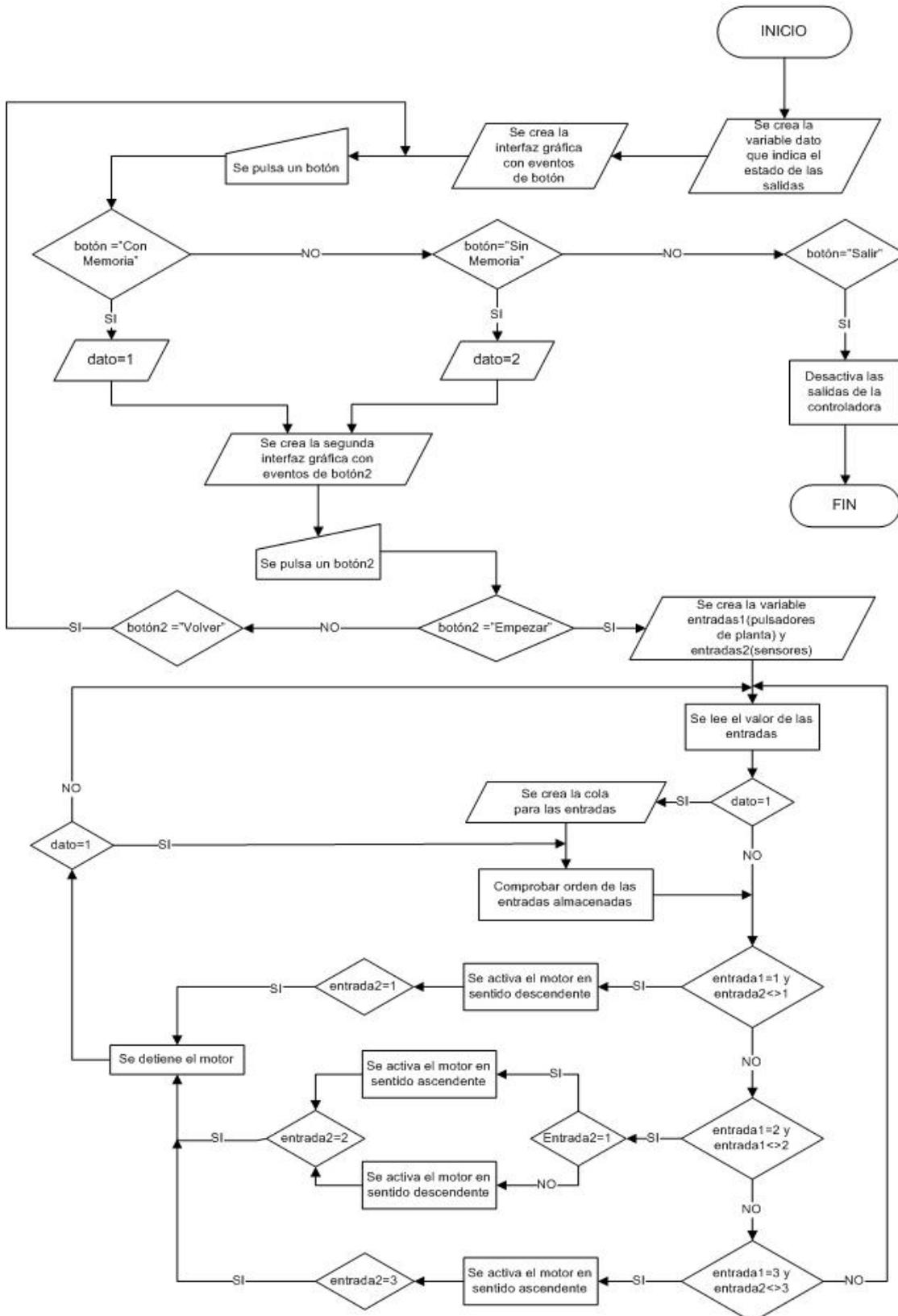
Interfaz de control de dispositivos externos por ordenador a través de puerto paralelo

Dato recibido en la entrada digital	Acción	Significado
1	Se ha presionado el pulsador de la planta 1	La cabina está entre plantas (1 + 0)
17	La cabina está en el piso 2 (1 + 16)	
33	La cabina está en el piso 3 (1 + 32)	
2	Se ha presionado el pulsador de la planta 2	La cabina está entre plantas (2 + 0)
10	La cabina está en el piso 1 (2 + 8)	
34	La cabina está en el piso 3 (2 + 32)	
4	Se ha presionado el pulsador de la planta 3	La cabina está entre plantas (4 + 0)
20	La cabina está en el piso 2 (4 + 16)	
12	La cabina está en el piso 1 (4 + 8)	
8	La cabina del ascensor ha pasado por el piso 1	La cabina está en el piso 1
10	La cabina está en el piso 1 y a la vez, se ha activado el pulsador 2 (8 + 2)	
12	La cabina está en el piso 1 y a la vez, se ha activado el pulsador 2 (8 + 4)	
16	La cabina del ascensor ha pasado por el piso 2	La cabina está en el piso 2
17	La cabina está en el piso 2 y a la vez, se ha activado el pulsador 1 (16 + 1)	
20	La cabina está en el piso 1 y a la vez, se ha activado el pulsador 3 (16 + 4)	
32	La cabina del ascensor ha pasado por el piso 2	La cabina está en el piso 3
33	La cabina está en el piso 3 y a la vez, se ha activado el pulsador 1 (32 + 1)	
34	La cabina está en el piso 3 y a la vez, se ha activado el pulsador 2 (32 + 2)	

PROGRAMACIÓN

Diagrama de flujo

ASCENSOR



Interfaz de control de dispositivos externos por ordenador a través de puerto paralelo

Programación en MSWLogo

La programación de este caso se estructura en los siguientes pasos:

1. Se crea un procedimiento para crear la ventana gráfica principal del programa. Para ello se utiliza la función **creaventana**. Dentro de la ventana se crean los botones con la función **creaboton**. Dentro de cada botón se establecerán entre los corchetes las funciones que se han de ejecutar una vez presionado el botón. Uno de los botones creados será el que se utiliza para salir de la aplicación, para lo cual se utiliza la orden **adios**.

```
creaventana " "Principal [ASCENSOR] 100 42 150 100 []  
creaboton "Principal "SinMemoria "Sin_Memoria 18 20 50 20 [SinMemo]  
creaboton "Principal "ConMemoria "Con_Memoria 80 20 50 20 [ConMemo]  
creaboton "Principal "Salir "Salir 50 50 50 20 [proc_salir adios]
```

En este procedimiento se crea y se inicializa a 0 una variable que llamaremos **bucle** que nos servirá para poder leer las entradas digitales de manera continuada. Además se crean e inicializan estas otras variables:

```
haz "destinocabina 0  
haz "situacioncabina -1  
haz "situacion_anterior 0  
haz "primeraiteracion "verdadero  
haz "lacabinasemueve "falso  
haz "micola []
```

donde:

- destinocabina: indica el piso al que va el ascensor
- situacioncabina: indica el piso donde se encuentra el ascensor
- situacion_anterior: indica el piso donde se encontraba anteriormente
- primeraiteracion: valor booleano que indica si es o no la primera iteración
- lacabinasemueve: indica si la cabina esta o no en movimiento
- micola: cola donde se almacenaran los pisos pulsados en el caso del funcionamiento con memoria.

2. Se crea un procedimiento para crear la ventana grafica para cada uno de los tipos de funcionamiento del ascensor, Con Memoria o Sin Memoria. Estas dos ventanas van a ser de igual aspecto, la única diferencia que existe es la manera de actuar el ascensor cuando se detectan entradas por los pulsadores de pisos. Se crean los botones de activar, desactivar el funcionamiento del ascensor o volver a la ventana inicial.

```
creaventana " "SinMemoria [ASCENSOR-Sin_Memoria] 100 42 150 100 []  
creaboton "SinMemoria "Activar "Activar 18 20 50 20 [control]  
creaboton "SinMemoria "Desactivar "Desactivar 80 20 50 20 [proc_parar]  
creaboton "SinMemoria "Volver "Volver 50 50 50 20 [proc_salir graficos]
```

3. El procedimiento control comprueba si la cabina está entre pisos, colocándola en este caso en el primer piso. Se comprueba si se ha pulsado o no algún botón de piso, haciendo subir o bajar la cabina del ascensor hasta el piso elegido, deteniéndose cuando se detecta el imán. Además se muestra la imagen del ascensor con la cabina en el piso que corresponde con la función **cargadib**.

```

para control
mientras [:bucle=0] [
    haz "entradas ve?
; se comprueba si la cabina esta entre pisos#####
    sisino (:entradas=0)[si (:primeraiteracion) [
        haz "destinocabina 1
        haz "primeraiteracion "falso
        bajar control]]
        [haz "primeraiteracion "falso]
; Se comprueba si se ha pulsado algun boton#####
    si O (O (:entradas=1) (:entradas=17)) (:entradas=33) [
        si Y NO(:situacioncabina=1) NO(:lacabinasemueve)[
            haz "destinocabina 1
            bajar control]]

    si O (O (:entradas=2) (:entradas=10)) (:entradas=34) [
        si Y NO(:situacioncabina=2) NO(:lacabinasemueve) [
            haz "destinocabina 2
            si (:situacioncabina=1) [subir control]
            si (:situacioncabina=3) [bajar control]
        ]
    ]

    si O (O (:entradas=4) (:entradas=20)) (:entradas=12) [
        si Y NO(:situacioncabina=3) NO(:lacabinasemueve)[
            haz "destinocabina 3
            subir control]]

; Ahora se controla cuando pasa por los imanes#####
    si (:entradas=8) [
        haz "situacioncabina 1
        si NO(:situacioncabina = :situacion_anterior)[
            haz "situacion_anterior 1
            cargadib "piso1.bmp]
        si (:destinocabina=1) [proc_parar control]]
    si (:entradas=16) [
        haz "situacioncabina 2
        si NO(:situacioncabina = :situacion_anterior)[
            haz "situacion_anterior 2
            cargadib "piso2.bmp]
        si (:destinocabina=2) [proc_parar control]]
    si (:entradas=32) [
        haz "situacioncabina 3
        si NO(:situacioncabina = :situacion_anterior)[
            haz "situacion_anterior 3
            cargadib "piso3.bmp]
        si (:destinocabina=3) [proc_parar control]]
    ]
fin

```

Interfaz de control de dispositivos externos por ordenador a través de puerto paralelo

4. El procedimiento control_Memo realiza las mismas funciones que el procedimiento anterior pero se almacena en una cola la serie de pisos detectados en las entradas con la función **cola**. Posteriormente se realizan los movimientos de la cabina en orden de salida de la cola gracias a la función **decola**.

```
para control_Memo
  mientras [:bucle=0] [
    haz "entradas ve?
;se comprueba si la cabina esta entre pisos#####
    sisino (:entradas=0)[si (:primeraiteracion) [
      haz "destinocabina 1
      haz "primeraiteracion "falso
      haz "lacabinasemueve "falso
      bajar ]]
      [haz "primeraiteracion "falso]
;Se comprueba si se ha pulsado algun boton#####
    si O (O (:entradas=1) (:entradas=17)) (:entradas=33) [
      si NO(:situacioncabina=1)[
        cola "micola 1
        Espera 10
      ]]
    si O (O (:entradas=2) (:entradas=10)) (:entradas=34) [
      si NO(:situacioncabina=2) [
        cola "micola 2
        espera 10
      ]]
    si O (O (:entradas=4) (:entradas=20)) (:entradas=12) [
      si NO(:situacioncabina=3)[
        cola "micola 3
        espera 10
      ]]
;Ahora se controla cuando pasa por los imanes#####
    si (:entradas=8) [
      haz "situacioncabina 1
      si NO(:situacioncabina = :situacion_anterior)[
        haz "situacion_anterior 1
        cargadib "piso1.bmp]
      si (:destinocabina=1) [haz "lacabinasemueve "falso proc_parar
        Espera 120 ]]
    si (:entradas=16) [
      haz "situacioncabina 2
      si NO(:situacioncabina = :situacion_anterior)[
        haz "situacion_anterior 2
        cargadib "piso2.bmp]
      si (:destinocabina=2) [haz "lacabinasemueve "falso proc_parar
        Espera 120 ]]
    si (:entradas=32) [
      haz "situacioncabina 3
      si NO(:situacioncabina = :situacion_anterior)[
        haz "situacion_anterior 3
        cargadib "piso3.bmp]
      si (:destinocabina=3) [haz "lacabinasemueve "falso proc_parar
        Espera 120 ]]
```

```

;Se sacan los valores de la cola para mover el ascensor
si (:lacabinasemueve="falso)[
haz "estado vacio? : micola
si (:estado = "falso) [
haz "frase [Cola no vacia]
muestra :frase
haz "cual decola "micola
si(:cual = 1)[haz "destinocabina 1 bajar]
si Y (:cual = 2)(:situacioncabina=1)[haz "destinocabina 2 subir]
si Y (:cual = 2)(:situacioncabina=3)[haz "destinocabina 2 bajar]
sí(:cual = 3)[haz "destinocabina 3 subir]
]
]
]
fin
    
```

5. Se crean los procedimientos para que el motor del ascensor suba o baje y para desactivar el motor. En estos procedimientos se va a utilizar la función M seguida del numero que indique el par de salidas digitales a activar, y tras esto se pondrán unas comillas con la letra D o I que indicara que el motor sube (activa la salida 1 y desactiva la salida 0) o que el motor baje (activa la salida 0 y desactiva la salida 1), con la letra P se le indica al motor que se detenga (desactiva las 2 salidas digitales).

```

para proc_parar
M1 "P
fin

para proc_salir
M1 "P
haz "lacabinasemueve "falso
haz "bucle 1
bt
Fin

para bajar
M1 "I
haz "lacabinasemueve "verdadero
Fin

para subir
M1 "D
haz "lacabinasemueve "verdadero
Fin
    
```

6. Se llama al procedimiento que crea la ventana gráfica fuera de cualquier procedimiento para que se cargue la aplicación gráfica nada más cargar el fichero de logo.

Interfaz de control de dispositivos externos por ordenador a través de puerto paralelo

Descargar el archivo programado en **MSWLogo**, descomprímalo y guárdelo en un directorio aparte. Contiene el fichero de código en MSWLogo (ascensor.lgo y las imágenes del ascensor).

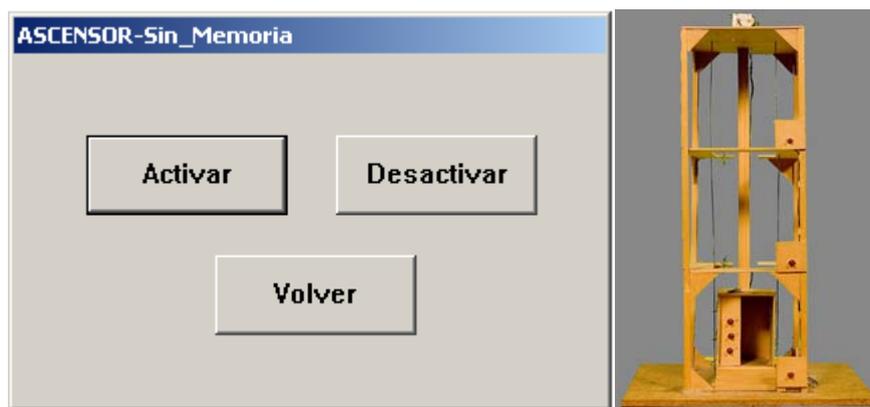
Ejecute el compilador **MSWLogo versión 6.5a** en castellano.

Vaya al menú del programa, Archivo/Abrir y seleccione el fichero **ascensor.lgo** que se descargó previamente.

Se visualizará la siguiente pantalla:



Independientemente del tipo de funcionamiento que se haya elegido, se mostrará la siguiente pantalla:



Programación en C

Colas.

Una cola es una estructura de datos donde los elementos se insertan por un extremo (final de la cola) y se suprimen por el otro (principio de la cola). Las colas se conocen también como FIFO First In, First Out (primero en entrar, primero en salir).

En C, para utilizar una cola, hay que realizar las siguientes primitivas:

- Crear
- Destruir
- Insertar elemento
- Quitar elemento
- Cola Vacía

Además, hay que definir una estructura de datos que se llamará Cola. Esto se hace la siguiente manera:

```
struct Cola{
    int Dato;
    struct Cola *Siguiete;
};
```

donde **Dato** es valor del elemento que queremos introducir en la cola y ***Siguiete** es un puntero que apuntara al elemento en la posición siguiente dentro de la cola.

Una vez definida la estructura Cola, se crea la cola con la que vamos a trabajar de la siguiente manera:

```
cola=NULL;
```

Definiendo la cola como NULL, lo que hacemos es crear una cola vacía.

Para introducir elementos en la cola se utiliza la primitiva Insertar:

```
Insertar(&Cabecera, &cola, elemento);
```

donde **Cabecera** indica que el **elemento** que se quiere almacenar ira en la primera posición si la **cola** esta vacía, sino, ira en el elemento siguiente al ultimo introducido.

La función Insertar es la siguiente:

```
void Insertar(struct Cola **Cabecera, struct Cola **cola, int valor)
{
    struct Cola *Nuevo;
    Nuevo=(struct Cola *)malloc(sizeof(int)); //Reserva dinámica de memoria
    if(Nuevo!=NULL){
        Nuevo->Dato=valor;
        Nuevo->Siguiete=NULL;
        if(EstaVacía(*Cabecera))
            *Cabecera=Nuevo;
        else
            (*cola)->Siguiete=Nuevo;
    }
    *cola=Nuevo;
}
```

Interfaz de control de dispositivos externos por ordenador a través de puerto paralelo

```
else MessageBox(0, "No hay espacio suficiente en la cola", "", 1);  
}
```

Como se puede observar, se comprueba si la cola esta o no vacía:

```
EstaVacía(*Cabecera);
```

donde ***Cabecera** es la posición que nos interesa, ya que si esta es NULL, significa que la cola esta vacía.

La función EstaVacía es la siguiente:

```
int EstaVacía(struct Cola *Cabecera)  
{  
    if(Cabecera==NULL)  
        return 1;  
    else  
        return 0;  
}
```

Para sacar elementos de la cola se utiliza la primitiva Borrar:

```
Borrar(&Cabecera, &cola);
```

donde **Cabecera** es el elemento que queremos eliminar de la **cola**.

La función Borrar es la siguiente:

```
void Borrar(struct Cola **Cabecera, struct Cola **cola)  
{  
    int valor;  
    struct Cola *Temporal;  
    valor=(*Cabecera)->Dato;  
    Temporal= *Cabecera;  
    *Cabecera=(*Cabecera)->Siguiete;  
    if(*Cabecera==NULL)  
        *cola=NULL;  
    free(Temporal);  
}
```

La programación de este caso se estructura en los siguientes pasos:

- 1.- Se crea un nuevo proyecto
- 2.- Se añaden al proyecto los archivos io.h, io.cpp, Primitivas_CNICE.CPP y Primitivas_CNICE.HPP y SDL.h (librería que permite añadir imágenes el la aplicación programada con C).
- 3.- Se crea el archivo main.c donde se incluirán las funciones necesarias para crear las ventanas
- 4.- Dentro del archivo main.c creado anteriormente hay que crear un hilo para que compruebe los sensores en todo momento y se añade la declaración a las funciones de la biblioteca io.dll de la siguiente manera:

```
#include "io.h"
```

También se añade la declaración a las funciones de la biblioteca SDL.dll de la siguiente manera:

```
#include <SDL.h>
```

5.- En nuestro archivo main.c se define una función que permite activar o desactivar las entradas digitales de la controladora y otra que permite leer el estado de las entradas digitales. Las funciones son las siguientes:

```
void encender (int led)
{
    LoadIODLL();
    PortOut(0x37A,0x7);
    PortOut(0x378,led);
}

int leedigital()
{
    int bajo=0;
    int alto=0;
    int d=0;

    LoadIODLL();
    PortOut(0x37A, 0x3);
    bajo = PortIn(0x379);
    bajo = (bajo & 0x78) / 8;
    bajo = ~bajo;
    bajo = bajo & 15;

    PortOut(0x37A, 0x1);
    alto = PortIn(0x379);
    alto = (alto & 0x78) / 8;
    alto = ~alto;
    alto = alto & 15;
    d = (alto * 16) | bajo;

    return d;
}
```

6.- Se crearán dos botones en nuestra ventana, uno para la ejecución "Sin Memoria" y otro para la ejecución "Con Memoria", y en cada uno de ellos se crea la ventana para manejar el ascensor. Se crean 3 botones: Activar, Desactivar y Volver.

```
switch(LOWORD(wParam))
{
    case 1: // botón Activar
        dato=0;
        break;
    case 2: // botón Desactivar
        motor = 0;
        dato=0;
        lacabinasemueve=0;
        encender(motor);
        break;
    case 3: // botón Salir
        motor = 0;

```

Interfaz de control de dispositivos externos por ordenador a través de puerto paralelo

```
        dato=0;
        lacabinasemueve=0;
        encender(motor);
        SendMessage(hwnd, WM_CLOSE, 0, 0);
        break;
    default:
        break;
}
```

En este caso es la ventana de la ejecución Sin Memoria. Se puede ver que se ha incluido lo siguiente:

```
    dato=0;
```

Esto determinará si la ejecución es Con o Sin Memoria.

7.- Se crea una función que realice la comprobación de los sensores. Esta función es la que ejecuta el hilo que hemos creado al principio. En ella se incluyen las llamadas a la función que lee el estado de los sensores, la función que enciende los motores y la función que muestra la imagen correspondiente. La función es la siguiente:

```
DWORD WINAPI Comprobar_Sensor(LPVOID parametros)
{
    int E, destinocabina;
    struct Cola *Cabecera, *cola;
    SDL_Surface *primera, *segunda, *tercera;
    SDL_Surface *screen;
    SDL_Rect destino;
    SDL_Rect rect;
    Cabecera=NULL;
    cola=NULL;

    for(;;){
        E = leedigital();
        switch (E)
        {
            case 1: case 17: case 33:
                if(dato==0){
                    if(lacabinasemueve==0){
                        destinocabina=1;
                        motor=1;
                        lacabinasemueve=1;
                        encender(motor);
                    }
                }
            else if (dato==1){
                Insertar(&Cabecera,&cola,1);
                Sleep(100);
            }
            break;
            case 2: case 10: case 34:
                if(dato==0){
                    if(lacabinasemueve==0){
                        if(destinocabina==1){
                            destinocabina=2;
                            motor=2;
                            lacabinasemueve=1;
                            encender(motor);
                        }
                        else if(destinocabina==3){
                            destinocabina=2;
                            motor=1;
                            lacabinasemueve=1;
                            encender(motor);
                        }
                    }
                }
            else if (dato==1){
                Insertar(&Cabecera,&cola,2);
                Sleep(100);
            }
            break;
        }
    }
}
```

```

case 4: case 20: case 12:
    if(dato==0){
        if(lacabinasemueve==0){
            destinocabina=3;
            motor=2;
            lacabinasemueve=1;
            encender(motor);
        }
    }else if (dato==1){
        Insertar(&Cabecera,&cola,3);
        Sleep(100);
    }
    break;
case 8:
    if(destinocabina==1){
        primera = SDL_LoadBMP("ascensor1.bmp");
        screen = SDL_SetVideoMode( 200, 308, 0, SDL_NOFRAME );
        if( screen == NULL ) {
            printf( "Error al entrar a modo grafico: %s\n", SDL_GetError() );
            SDL_Quit();
        }
        rect = (SDL_Rect) {0, 0, 200, 200};
        SDL_BlitSurface(primera, NULL,screen,&rect);
        SDL_Flip(screen);
        encender(0);
        lacabinasemueve=0;
    }
    break;
case 16:
    if(destinocabina==2){
        segunda = SDL_LoadBMP("ascensor2.bmp");
        screen = SDL_SetVideoMode( 200, 308, 0, SDL_NOFRAME );
        if( screen == NULL ) {
            printf( "Error al entrar a modo grafico: %s\n", SDL_GetError() );
            SDL_Quit();
        }
        rect = (SDL_Rect) {0, 0, 200, 200};
        SDL_BlitSurface(segunda, NULL,screen,&rect);
        SDL_Flip(screen);
        encender(0);
        lacabinasemueve=0;
    }
    break;
case 32:
    if(destinocabina==3){
        tercera = SDL_LoadBMP("ascensor3.bmp");
        screen = SDL_SetVideoMode( 200, 308, 0, SDL_NOFRAME );
        if( screen == NULL ) {
            printf( "Error al entrar a modo grafico: %s\n", SDL_GetError() );
            SDL_Quit();
        }
        rect = (SDL_Rect) {0, 0, 200, 200};
        SDL_BlitSurface(tercera, NULL,screen,&rect);
        SDL_Flip(screen);
        encender(0);
        lacabinasemueve=0;
    }
    break;
default:
    break;
}

if(dato==1){
    if(!EstaVacia(Cabecera)){
        if(lacabinasemueve==0){
            switch (Cabecera->Dato){
                case 1:
                    Borrar(&Cabecera,&cola);

```

Interfaz de control de dispositivos externos por ordenador a través de puerto paralelo

```
        Sleep(1000);
        destinocabina=1;
        lacabinasemueve=1;
        encender(1);
        break;
    case 2:
        Borrar(&Cabecera,&cola);
        Sleep(1000);
        if(destinocabina==1){
            destinocabina=2;
            lacabinasemueve=1;
            encender(2);
        }
        else if(destinocabina==3){
            destinocabina=2;
            lacabinasemueve=1;
            encender(1);
        }
        break;
    case 3:
        Borrar(&Cabecera,&cola);
        Sleep(1000);
        destinocabina=3;
        lacabinasemueve=1;
        encender(2);
        break;
    default:
        break;
}
}}}}}
```

Como se puede observar aquí entra en juego la variable dato. Si es 1, significa que el ascensor tiene memoria, y si es 0 no posee esa memoria. Para la ejecución del ascensor con memoria, se crea una cola en la que se almacena la secuencia de los botones pulsados. En cada piso esperará un tiempo determinado.

Explicación de las funciones de la librería SDL.

Cada vez que se quiera mostrar una imagen se indicara en nuestro proyecto lo siguiente:

```
foto = SDL_LoadBMP("imagen.bmp");
screen = SDL_SetVideoMode(200, 308, 0, SDL_NOFRAME );
if( screen == NULL ) {
    printf( "Error al entrar a modo grafico: %s\n", SDL_GetError() );
    SDL_Quit();
    return -1;
}
rect.x=0;
rect.y=0;
rect.w=primera1->w;
rect.h=primera1->h;
destino.x=0;
destino.y=0;
SDL_BlitSurface(primera1, &rect, screen, &destino);
SDL_Flip(screen);
```

donde **foto** y **screen** son del tipo `SDL_Surface` y **rect** y **destino** es del tipo `SDL_Rect`.

- `SDL_LoadBMP`: carga la imagen .bmp que queramos
- `SDL_SetVideoMode` (int width, int height, int bpp, Uint32 flags): configure un modo de video con una anchura (width), una altura (height) y unos bits-por-pixeles. El parámetro flags indica el tipo de ventana que se quiere. En nuestro caso una ventana sin titulo no borde.
- `SDL_BlitSurface`(imagen, &rect, screen, &destino): pega desde la imagen, la porción seleccionada por rect sobre la superficie screen en el destino indicado por destino.
- `SDL_Flip`(screen): muestra la imagen que se ha seleccionado.

8.- Una vez creados los botones con la función que les corresponden, se compila comprobando que no hay ningún error.

9.- Una vez que se ha comprobado que no hay ningún error en nuestro código, se ejecuta y se comprueba el funcionamiento de la barrera. Al ejecutar el proyecto se creará el fichero **Ascensor.exe**

Descargue los diferentes archivos que forman todo el proyecto programado en C, descomprímalos y guárdelos en un directorio aparte. Ejecute el fichero **Ascensor.exe**. Se visualizará la siguiente pantalla:



Se pulsa en la opción que se desee y aparecerá la siguiente pantalla (es común para las dos opciones):



Pulsando los botones que se le presentan podrá Activar y Desactivar el ascensor o Volver a la pantalla anterior para elegir otra opción de la aplicación.

Nota:

En la aplicación programada con C, la imagen puede no aparecer al lado de la ventana. En este caso basta con mover nuestra ventana y se verá correctamente la imagen.

ALGUNAS FOTOS DE LA ESTRUCTURA DEL ASCENSOR

