

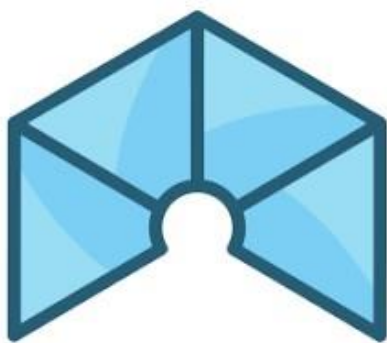


---

# COLECCIÓN DE EJERCICIOS

---

Alhambra II



## Contenido

Tema 8 .....	3
Ejercicio 8.1. ....	3
Ejercicio 8.2. ....	7
Ejercicio 8.3. ....	9
Tema 9 .....	12
Ejercicio 9.1. ....	12
Ejercicio 9.2. ....	12
Ejercicio 9.3. ....	12
Ejercicio 9.4. ....	12
Tema 10 .....	13
Ejercicio 10.1. ....	13
Ejercicio 10.2. ....	13
Ejercicio 10.3. ....	13
Ejercicio 10.4. ....	13
Tema 11 .....	14
Ejercicio 11.1. ....	14
Tema 12 .....	14
Ejercicio 12.1. ....	14
Ejercicio 12.2. ....	14
Tema 13 .....	15
Ejercicio 13.1. ....	15
Tema 15 .....	15
Ejercicio 15.1. ....	15
Ejercicio 15.2. ....	15
Ejercicio 15.3. ....	15
Tema 16 .....	16
Ejercicio 16.1. ....	16
Ejercicio 16.2. ....	16
Ejercicio 16.3. ....	16
Tema 17 .....	17
Ejercicio 17.1. ....	17
Ejercicio 17.2. ....	17
Tema 18 .....	17

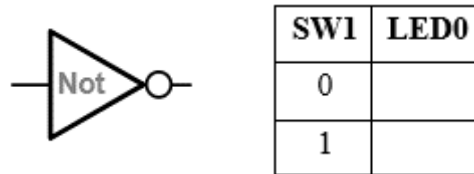
Ejercicio 18.1.....	17
Ejercicio 18.2.....	17
Tema 20 .....	18
Ejercicio 20.1.....	18
Ejercicio 20.2.....	18
Tema 21 .....	18
Ejercicio 21.1.....	18
Ejercicio 21.2.....	18
Tema 22 .....	19
Ejercicio 22.1.....	19
Ejercicio 22.2.....	19
Ejercicio 22.3.....	19
Tema 23 .....	20
Ejercicio 23.1.....	20
Ejercicio 23.2.....	20
Ejercicio 23.3.....	20
Tema 24 .....	21
Ejercicio 24.1.....	21
Ejercicio 24.2.....	21
Tema 25 .....	22
Ejercicio 25.1.....	22
Tema 26 .....	23
Ejercicio 26.1.....	23
Ejercicio 26.2.....	23

## Tema 8

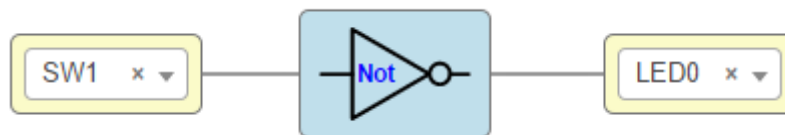
### Ejercicio 8.1.

Rellenar la tabla de la verdad de las puertas lógicas básicas y verificar en Icestudio.

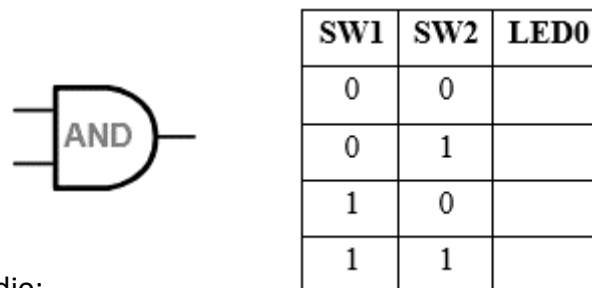
#### a) NOT



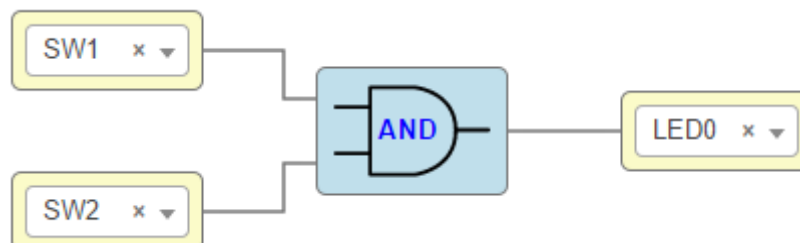
Circuito en Icestudio:



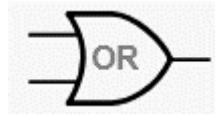
#### b) AND



Circuito en Icestudio:

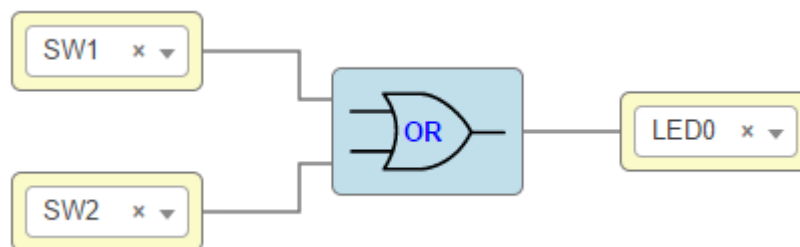


**c) OR**



SW1	SW2	LED0
0	0	
0	1	
1	0	
1	1	

Circuito en Icestudio:

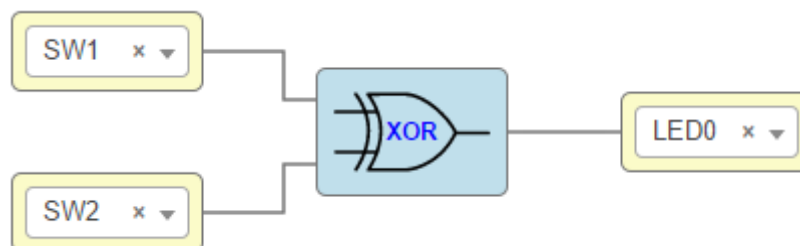


**d) XOR**

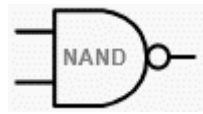


SW1	SW2	LED0
0	0	
0	1	
1	0	
1	1	

Circuito en Icestudio:

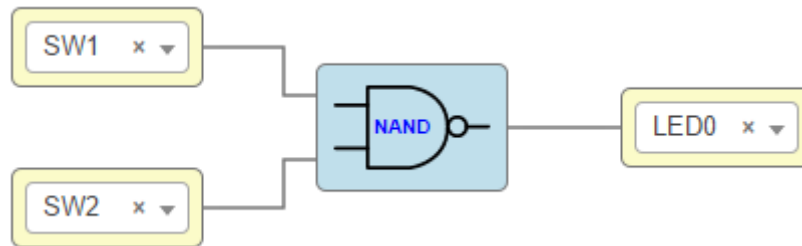


**e) NAND**



SW1	SW2	LED0
0	0	
0	1	
1	0	
1	1	

Circuito en Icestudio:

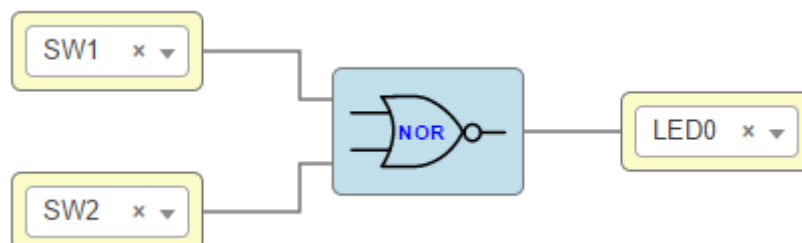


**f) NOR**



SW1	SW2	LED0
0	0	
0	1	
1	0	
1	1	

Circuito en Icestudio:

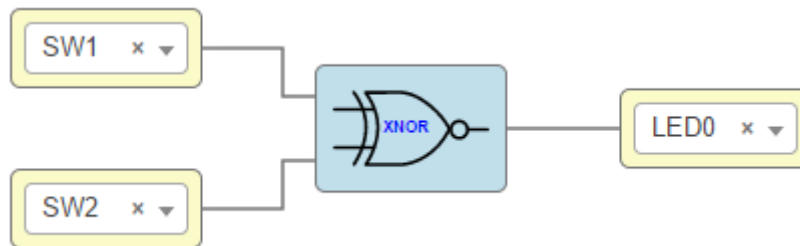


g) XNOR



SW1	SW2	LED0
0	0	
0	1	
1	0	
1	1	

Circuito en Icestudio:



Resultados:

a)

SW1	LED0
0	1
1	0

b)

SW1	SW2	LED0
0	0	0
0	1	0
1	0	0
1	1	1

c)

SW1	SW2	LED0
0	0	0
0	1	1
1	0	1
1	1	1

d)

SW1	SW2	LED0
0	0	0
0	1	1
1	0	1
1	1	0

e)

SW1	SW2	LED0
0	0	1
0	1	1
1	0	1
1	1	0

f)

SW1	SW2	LED0
0	0	1
0	1	0
1	0	0
1	1	0

g)

SW1	SW2	LED0
0	0	1
0	1	0
1	0	0
1	1	1

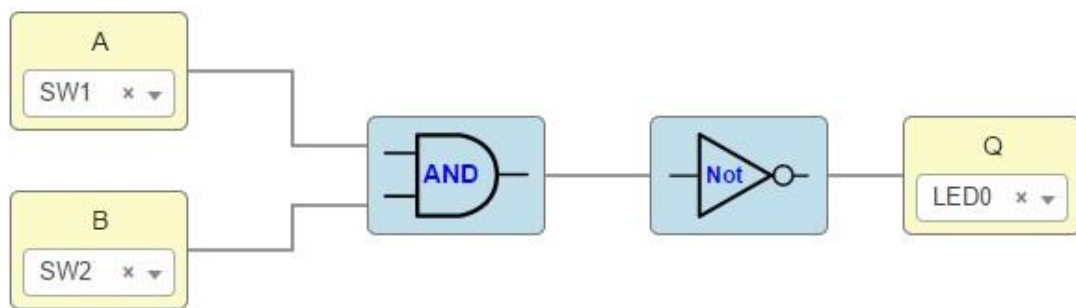
## Ejercicio 8.2.

Dibujar las siguientes puertas utilizando solamente las puertas lógicas AND, OR y NOT.

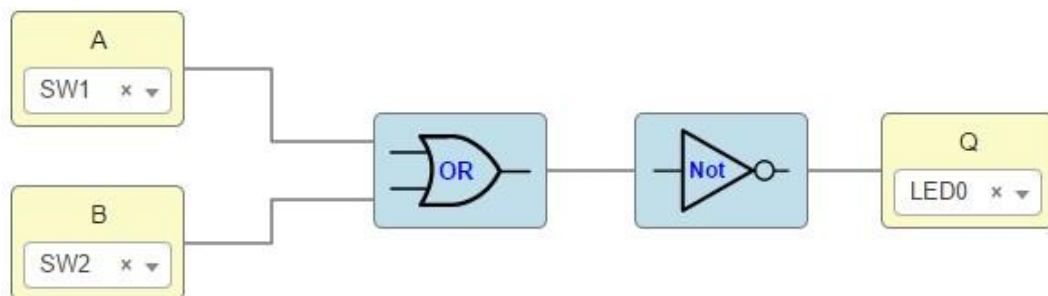
- a) NAND
- b) NOR
- c) XOR
- d) XNOR

### Resolución

- a) NAND

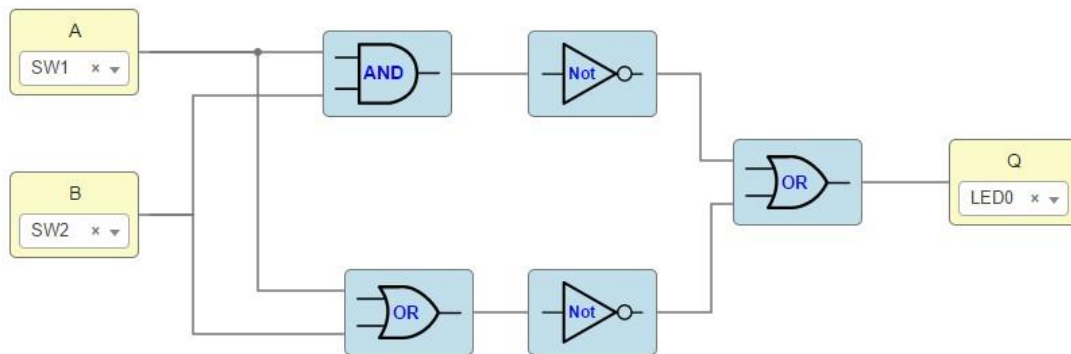


- b) NOR

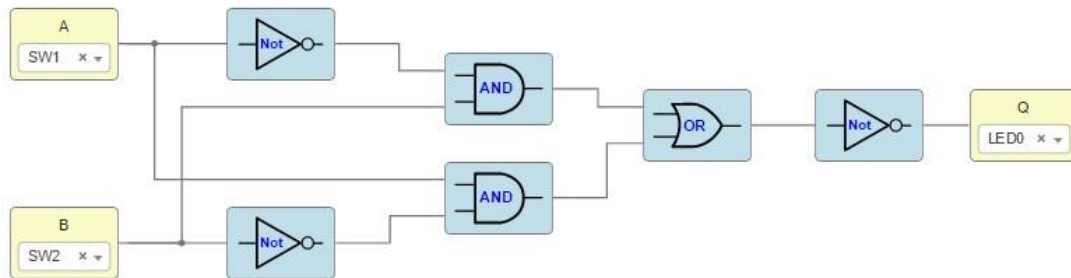




c) XOR



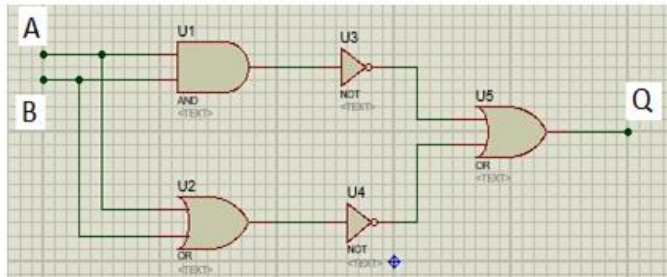
d) XNOR



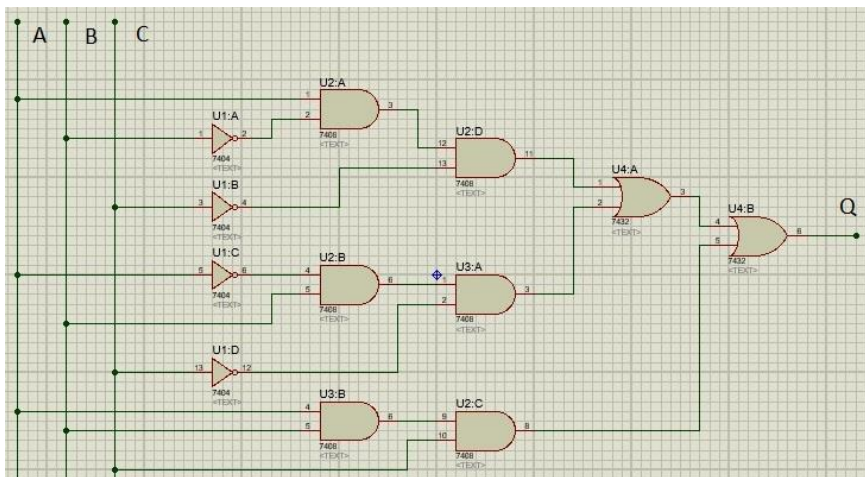
### Ejercicio 8.3.

Validar los siguientes circuitos digitales en Icestudio y rellenar las tablas de la verdad.

a)



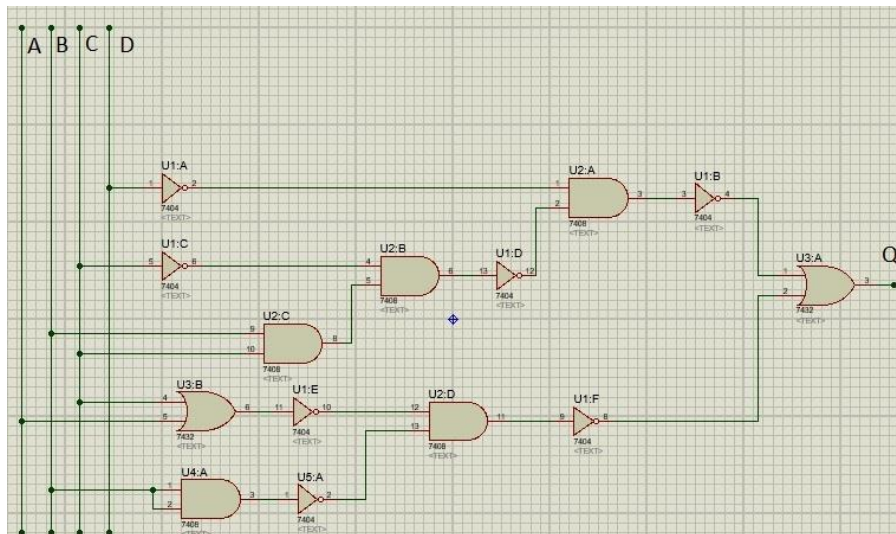
b)



A	B	C	Q
0	0	0	
0	0	1	
0	1	0	
0	1	1	
1	0	0	
1	0	1	
1	1	0	
1	1	1	

Para resolver el siguiente circuito en la FPGA conectaremos las tres entradas (3 interruptores) en tres pines digitales de la FPGA, en nuestro caso en D1, D2 y D3. La salida (mediante un led) la conectaremos al pin digital D13.

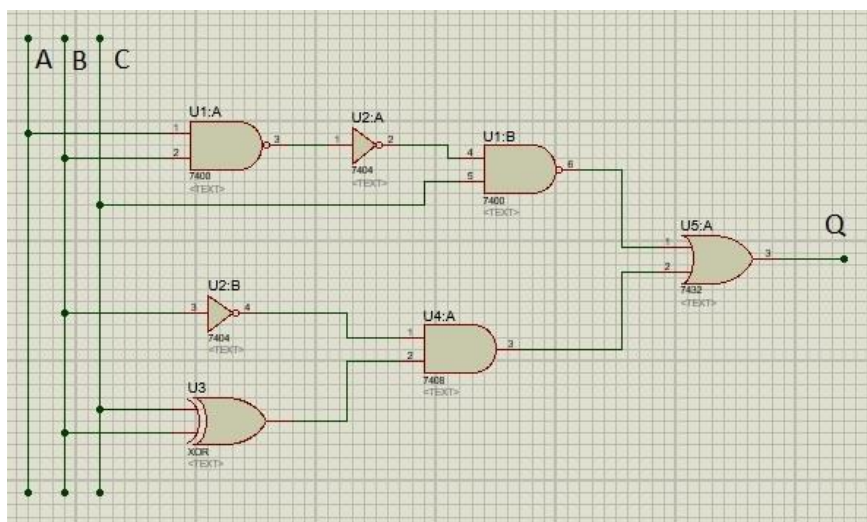
c)



A	B	C	D	Q
0	0	0	0	
0	0	0	1	
0	0	1	0	
0	0	1	1	
0	1	0	0	
0	1	0	1	
0	1	1	0	
0	1	1	1	
1	0	0	0	
1	0	0	1	
1	0	1	0	
1	0	1	1	
1	1	0	0	
1	1	0	1	
1	1	1	0	
1	1	1	1	

Para resolver el siguiente circuito en la FPGA conectaremos las cuatro entradas (4 interruptores) en cuatro pines digitales de la FPGA, en nuestro caso en D1, D2, D3 y D4. La salida (mediante un led) la conectaremos al pin digital D13.

d)



A	B	C	Q
0	0	0	
0	0	1	
0	1	0	
0	1	1	
1	0	0	
1	0	1	
1	1	0	
1	1	1	

Para resolver el siguiente circuito en la FPGA conectaremos las tres entradas (3 interruptores) en tres pines digitales de la FPGA, en nuestro caso en D1, D2 y D3. La salida (mediante un led) la conectaremos al pin digital D13.

### Resolución

A	B	Q
0	0	1
0	1	1
1	0	1
1	1	0

A	B	C	Q
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

A	B	C	D	Q
0	0	0	0	0
0	0	0	1	1
0	0	1	0	1
0	0	1	1	1
0	1	0	0	1
0	1	0	1	1
0	1	1	0	1
0	1	1	1	1
1	0	0	0	1
1	0	0	1	1
1	0	1	0	1
1	0	1	1	1
1	1	0	0	1
1	1	0	1	1
1	1	1	0	1
1	1	1	1	1

A	B	C	Q
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0

## Tema 9

---

### Ejercicio 9.1.

Hacer un circuito que haga parpadear los LEDs 0 y 7 a la vez, a la velocidad de un parpadeo por segundo.

### Ejercicio 9.2.

Hacer un circuito que haga parpadear todos los LEDs de la Alhambra II. Los pares parpadearan a 4Hz y los impares a 2Hz.

### Ejercicio 9.3.

Hacer un circuito que haga parpadear solo algunos LEDs de la Alhambra. Cada uno a una velocidad diferente:

- LED 6 → 10Hz
- LED 4 → 7Hz
- LED 2 → 4Hz
- LED 0 → 1Hz

### Ejercicio 9.4.

Hacer un circuito que haga parpadear dos LEDs externos. Cada uno a velocidad diferente.

## Tema 10

---

### Ejercicio 10.1.

Mover dos servos diferentes en fase usando la misma señal de control.

### Ejercicio 10.2.

Mover dos servos a posiciones contrarias con la misma señal de control.

### Ejercicio 10.3.

Mover dos servos diferentes con diferentes señales de control.

### Ejercicio 10.4.

**Apertura de una cerradura.** Utilizando 3 interruptores introduciremos un código de 3 bits (101). Veremos el estado de los interruptores en 3 LEDs diferentes cualesquiera. Cuando se introduzca el código correcto, el servo cambiará de posición, simulando la apertura de la puerta.

## Tema 11

---

### Ejercicio 11.1.

**Limpiaparabrisas a dos velocidades.** Usando un servo para simularlo, le pondremos dos velocidades, controladas por un pulsador de la Alhambra. El modo lento será de 1Hz, mientras que el modo rápido será de 4Hz.

## Tema 12

---

### Ejercicio 12.1.

Programar el robot para que vaya siempre hacia adelante al pulsar un botón.

### Ejercicio 12.2.

En este ejercicio el movimiento será automático, estará 1 segundo parado y otro segundo en marcha. Además, el movimiento será en el sentido contrario al anterior.

## Tema 13

---

### Ejercicio 13.1.

Hacer un robot que siga un objeto, usando dos sensores, uno para cada rueda.

## Tema 15

---

### Ejercicio 15.1.

Diseñar un comparador de 2 bits. Esto es, mediante dos entradas, escribiremos un número, y con otras dos entradas otro, y la salida se activará solamente cuando ambos números serán iguales

### Ejercicio 15.2.

**Apertura de caja fuerte.** Diseñar un circuito usando la tabla de la verdad para que al introducir un código de 3 bits se abra la caja fuerte. Introduciremos el código mediante 3 interruptores y simularemos la apertura de la caja mediante un servo.

### Ejercicio 15.3.

**Detector de paridad.** Diseñar un circuito de 4 bits de entrada, usando la tabla de la verdad, que detecte cuando hay un número par de bits. Se encenderá un LED cuando la condición se cumpla. La entrada 0000 es **par**.



## Tema 16

---

### Ejercicio 16.1.

Diseñar un circuito digital que muestre por los **8 LEDs** de la Alhambra la siguiente **secuencia**: 0, 15, 255 y 15, que se repetirá indefinidamente. Utilizar un multiplexor de 4 a 1 de 8 bits.

### Ejercicio 16.2.

Diseñar un circuito digital que saque por los **LEDs 7 y 6** los pulsadores **SW2** y **SW1**, usando un **bus de 2-bits**. Además, se sacará por el **LED 0** el resultado de la **operación AND** entre SW1 y SW2, obteniendo sus valores del bus de 2 bits, mediante un separador.

### Ejercicio 16.3.

Diseñar un circuito digital que **cifre un número de 4-bits** mediante el **intercambio** de sus **bits centrales** (Bits 1 y 2), dejando el de mayor y menor peso iguales (bit 3 y bit 0). El número se introducirá mediante **4 interruptores externos** que entran en la Alhambra mediante un **bus de 4 bits**. El número sin cifrar se sacará por los LEDs 7,6,5 y 4, y el número cifrado por los LEDs 3,2,1 y 0

## Tema 17

---

### Ejercicio 17.1.

Diseñar una puerta OR de 3 entradas usando puertas OR de 2 entradas.

### Ejercicio 17.2.

Comparador de números de 2 bits. Insertaremos un valor como parámetro y se comparará con el valor dado por dos interruptores. Cuando los valores coincidan, se encenderá un LED de la Alhambra.

## Tema 18

---

### Ejercicio 18.1.

Utilizando un circuito combinacional de 2 entradas y 8 salidas, diseñar un circuito que muestre por los LEDs una secuencia de 4 estados que sea: LEDs 0, 1 y 2  $\rightarrow$  3 y 4  $\rightarrow$  5, 6, 7  $\rightarrow$  3 y 4. El proceso se repite a una frecuencia de 2 Hz.

### Ejercicio 18.2.

Construir un circuito de 2 entradas y 2 salidas, que, siendo sus entradas dos sensores infrarrojos, tenga el siguiente comportamiento:

IR izquierdo	IR derecho	Movimiento del icebot
No detecta	No detecta	Parado
No detecta	Detecta	Giro izquierda (Arco)
Detecta	No detecta	Giro derecha (arco)
Detecta	Detecta	Adelante

## Tema 20

---

### Ejercicio 20.1.

Diseñar un circuito que realice un conteo de 0 a 7 automáticamente usando corazones y que se visualice en un display.

### Ejercicio 20.2.

Diseñar un conversor binario a decimal con 4 entradas. Podremos visualizar dígitos del 0 al 9 y letras de la A a la F en el display.

## Tema 21

---

### Ejercicio 21.1.

**Notificación de presencia.** Diseñar un circuito que notifique que un sensor infrarrojo ha detectado un objeto. Se usará un LED como salida.

### Ejercicio 21.2.

**Apertura caja fuerte.** Diseñar un sistema de seguridad para una caja fuerte. El código será de 3 bits. Una vez correcto, se deberá pulsar un botón para que la puerta se abra. Al abrirse, se apagará un LED rojo (que estará encendido mientras esté cerrada) y se encenderá un LED verde.

## Tema 22

---

### Ejercicio 22.1.

Diseñar un circuito que active una salida (LED) mediante un tic cuando el sensor infrarrojo detecte un objeto.

### Ejercicio 22.2.

Diseñar un circuito que, usando un bucle infinito, haga 3 tareas diferentes:

- Primero, durante 3 segundos, un LED parpadeará a una frecuencia de 2Hz
- Después, se “abrirá una barrera” durante 4 segundos.
- Al cerrarse la barrera, se encenderá un Led Rojo durante 2 segundos

### Ejercicio 22.3.

Diseñar un circuito que varíe la posición de un servo entre dos posiciones pulsando un botón, usando una señal PWM creada con un corazón y temporizadores.

## **Tema 23**

---

### **Ejercicio 23.1.**

Diseñar un circuito que cuente hasta 9 pulsaciones y se visualice en un display 7 segmentos.

### **Ejercicio 23.2.**

Diseñar un contador de 4 bits a partir de dos de 2 bits, mostrando una cuenta en hexadecimal. Se muestra la cuenta del primer contador en dos LEDs

### **Ejercicio 23.3.**

Diseñar un circuito con el que se visualice un HOLA en un solo display automáticamente. Las letras tendrán que aparecer secuencialmente. Además, se podrá elegir entre velocidad lenta o velocidad rápida.

## Tema 24

---

### Ejercicio 24.1.

Diseñar un circuito para abrir la puerta de una caja fuerte. Primero se introduce un código de 3 bits, usando biestables D, y cuando se pulse la tecla “Enter”, se comprueba si es correcto. En caso de que lo sea, se moverá un servo. Para cerrar la caja habrá que introducir cualquier código incorrecto y pulsar “Enter”.

### Ejercicio 24.2.

Diseñar un circuito para hacer multiplicaciones por dos. Empezaremos con un 1 y se multiplicará por 2 cada vez que se pulse un botón, hasta el 8. Visualizaremos el número en decimal mediante un display.

## Tema 25

---

### Ejercicio 25.1.

Diseñar una consola de control básica, desde donde controlaremos una alarma luminosa, una puerta accionada por un servo y un motor. Esto lo haremos introduciendo un código de 3 bits mediante 3 interruptores externos y un pulsador ENTER. Veremos el código seleccionado mediante un display. Estas serán las acciones a programar:

Comando	Código	Descripción
RESET	0	Apagar todos los dispositivos (volver al estado inicial)
AlarmON	1	Activar Alarma luminosa
AlarmOFF	2	Desactivar Alarma luminosa
OPEN	3	Abrir la puerta (Se mueve el servo)
CLOSE	4	Cerrar la puerta (Se mueve el servo)
MotorON	5	Encender el motor
MotorOFF	6	Apagar motor
MotorDIR	7	Cambiar el sentido de la marcha del motor

## Tema 26

---

### Ejercicio 26.1.

Diseñar un circuito con el que controlar un servo desde un APP en el móvil usando como base el ejemplo de la teoría.

### Ejercicio 26.2.

Diseñar un circuito para contar las veces que un sensor infrarrojo detecta un objeto. El dato enviado constará de los dos dígitos (unidades y decenas) y un fin de línea (\n). Este dato se visualizará en cualquier terminal (PC o Móvil)