

SIMULACIÓN Y RESOLUCIÓN DE PROBLEMAS IMPLEMENTANDO CIRCUITOS COMBINACIONALES

1.PLANTEAMIENTO DEL PROBLEMA

Se desconoce el funcionamiento, diseño y conexión de un circuito votador, un circuito comparador y un circuito con representación binaria, para lo cual se formularon las siguientes preguntas:

- ¿Cómo se diseña un circuito votador?
- ¿Cómo se diseña un circuito comparador?
- ¿Cómo se diseña un circuito que tenga como entradas la representación binaria de la hora actual menos ocho?

2.OBJETIVOS

Objetivo general

Realizar el diseño de diferentes circuitos combinacionales con aplicaciones.

Objetivos específicos

- Diseñar un circuito votador y la influencia en la seguridad de aviones modernos.
- Diseñar un circuito que tenga como entradas la representación binaria aplicado a un horario laboral de una factoría.
- Diseñar un circuito comparador exclusivamente con puertas NOR.

3.ESTADO DEL ARTE

En 2009, Zhaojun Gan, Gang Shi y Tao Shang de la Facultad de Ciencias de la Información e Ingeniería, Universidad de Ciencia y Tecnología de Wuhan realizaron un estudio implementando circuitos lógicos combinacionales en el diseño automático de la programación genética (GP). GP usando una estructura de árbol o una estructura lineal en lugar de cadenas para representar un circuito lógico combinacional. En el método GP, los nodos hoja y los nodos padres representan señales de entrada y puertas lógicas en circuitos, respectivamente. Los circuitos también evolucionan con algoritmos de selección clonal basado en gráficos (Zhaojun Gan, 2009, p.1) [1].

Alidoust Zahra y Basiri Mohammad, del Departamento de ingeniería informática Departamento de Ingeniería Universidad de Isfahan y de la Universidad Shahrood, Irán en 2017 centraron su estudio en mejorar el algoritmo genético a través de la agrupación para el diseño de circuitos lógicos combinacionales, los diseños evolutivos son buenas alternativas para el diseño de circuitos lógicos pero tienen un inconveniente común, a saber, la alta aleatoriedad de su método cruzado. Para superar este inconveniente el método propuesto, se adopta el algoritmo k-means para optimizar el algoritmo genético con el fin de aumentar la eficiencia y reducir el costo de producción. Los criterios de optimización de los elementos del circuito, como el recuento de puertas de los transistores y el consumo de energía (Zahra Alidousti, 2017, p.1) [2].

Mukherjee, B. y Dandapat, Dept, Jadavpur University Kolkata, en el Simposio internacional 2010 presentaron un estudio sobre el diseño de sistemas electrónicos *mediante el método combinacional cíclico para VLSI de baja potencia*. El objetivo principal del circuito cíclico es introducir retroalimentación estructural y evitar la retroalimentación lógica para obtener una salida primaria combinatoria. El estudio del circuito cíclico incluye análisis funcional, es decir, para determinar qué valores aparecerán, el análisis de tiempo que determina cuándo

aparecerán estos valores y el área. El objetivo del estudio es diseñar y verificar diferentes circuitos combinacionales en un método cíclico mediante la optimización del área, la potencia y el retraso.

(L Mukherjee, B. y Dandapat, 2010, p.1) [3].

Youjun Xu, Dantong Ouyang1, Yuxin Ye, y Jialiang He, Facultad de Informática y Tecnología Jilin University, Changchun, China, en su trabajo de investigación basado en la *solución de problemas SAT con álgebra booleana expusieron que en* las últimas décadas, se proponen muchos métodos de SAT, basado en la resolución, por medio de la regla de extensión, en el estudio de la regla de extensión, encontramos que el problema SAT se puede resolver con algoritmos de configuración de golpe. Si podemos encontrar un conjunto de aciertos de un conjunto de cláusulas, y si no hay ningún par de literales complementarios en el conjunto de aciertos, el conjunto de cláusulas es satisfactoria. El algoritmo BHS basado en el álgebra booleana propuesto por Jiang es un algoritmo de conjunto de golpes eficiente. (Xu, Y., Ouyang, D., Ye, 2010, p.1) [4].

Ruanqianqian Huang y Franklyn Turbak del Wellesley College, Department of Computer Science en 2019 implementaron un diseño para la conversión bidireccional entre bloques y texto para App Inventor mediante fragmentos de código visual que evitan errores semánticos sintácticos y estáticos y reducen la carga cognitiva para este fin diseñaron un sistema de modo dual para MIT App Inventor que admite representaciones textuales para bloques, espacios de trabajo, pantallas y proyectos completos que permiten la conversión bidireccional entre bloques isomorfos y representaciones de texto, permitiendo que individuos de varios niveles de experiencia en programación se relacionen con la interfaz (R. Huang,F. Turbak,2019)[5].

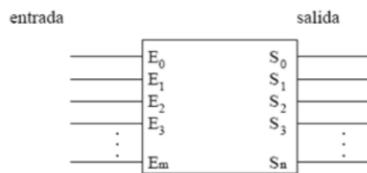
Para el Producto de Unidad presente, se utilizó una estructura de árbol o una estructura lineal en lugar de cadenas para representar un circuito lógico combinacional (Zhaohui Gan, 2009, p.1), para la optimización de los elementos del circuito, como el recuento de puertas de los transistores y el consumo de energía se utilizó parte de los postulados empleados en algoritmo genético con circuitos combinacionales (Zahra Alidousti,2017, p.1) el entorno de programación para el aplicativo se hizo sobre la base de MIT App Inventor, el sistema tiene la capacidad de proyectar un número decimal de salida en la pantalla del dispositivo android simulando la proyección de un display de 7 segmentos (R. Huang,F. Turbak,2019).

4.MARCO TEÓRICO

Circuito combinacional

Un circuito combinacional es un circuito electrónico, en el que el valor de sus salidas en un determinado instante, dependen del valor de las entradas en ese mismo instante. Es decir, es un circuito que carece de memoria. Trabajan con números, y con la tecnología con la que están realizados, estos números están representados en binario.

El siguiente circuito:



En un determinado instante van a depender de las entradas en ese preciso instante. Estos circuitos se caracterizan porque no almacenan información. Las salidas están relacionadas con las entradas a través de una función booleana.

Cada bit de salida de un circuito combinacional, se obtiene mediante una función booleana aplicado a las variables de entrada. Así, si un circuito tiene n salidas, necesitaremos n funciones booleanas para caracterizarlo.

Así, un circuito combinacional que tiene 3 entradas: A, B y C, y dos salidas F, G, que son dos funciones booleanas que dependen de las variables de entrada: F(A,B,C) y G(A,B,C), siendo por ejemplo, el valor de las funciones:

$$F = A + C'D$$

$$G = AB' + D$$

Se obtiene un circuito combinacional como este:



Circuito comparador

Un circuito comparador compara dos entradas binarias (A y B de n bits) para indicar la relación de igualdad o desigualdad entre ellas por medio de "tres banderas lógicas" que corresponden a las relaciones A igual B, A mayor que B y A menor que B. Cada una de estas banderas se activará solo cuando la relación a la que corresponde sea verdadera, es decir, su salida será 1 y las otras dos producirán una salida igual a cero.

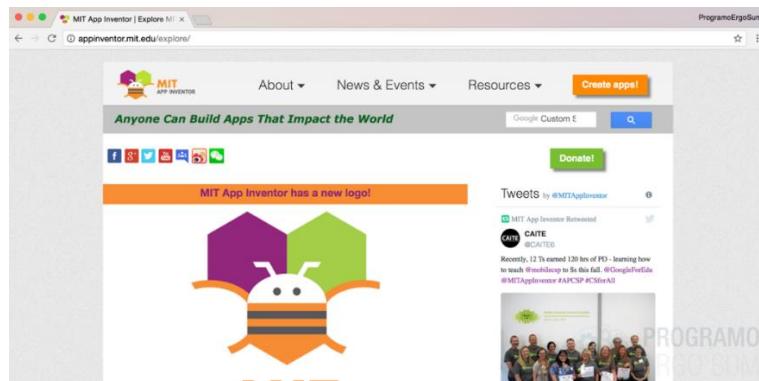
Circuito votador

Es conocido el uso de redundancia en aplicaciones electrónicas con altos requisitos de seguridad ante fallos. Las técnicas de redundancia implican utilizar una pluralidad de circuitos independientes para obtener un solo resultado. Cada uno de los circuitos genera una señal de salida, por lo que es necesario un sistema votador ("voter" en inglés), que indica cual es el valor de salida más repetido. Dicho valor más repetido pasa a considerarse el resultado de salida del sistema de redundancia.

App Inventor

App Inventor es un entorno de desarrollo de software creado por Google para la elaboración de aplicaciones destinadas al sistema operativo de Android. El lenguaje es gratuito y se puede acceder fácilmente de la web. Las aplicaciones creadas con App Inventor están limitadas por su simplicidad, aunque permiten cubrir un gran número de necesidades básicas en un dispositivo móvil.

Con App Inventor, se espera un incremento importante en el número de aplicaciones para Android debido a dos grandes factores: la simplicidad de uso, que facilitará la aparición de un gran número de nuevas aplicaciones; y Google Play, el centro de distribución de aplicaciones para Android donde cualquier usuario puede distribuir sus creaciones libremente.

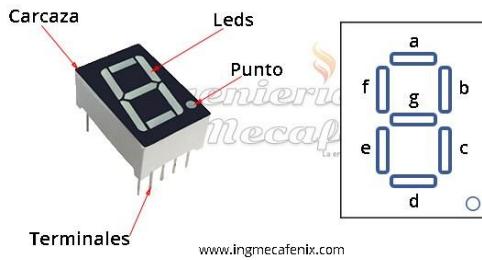


Display De 7 Segmentos

El display de 7 segmentos es un dispositivo electrónico que se utiliza para representar visualmente números y algunos caracteres. Este display es muy popular debido a su gran efectividad y simplicidad al momento de utilizarlo.

Partes de un display de 7 segmentos

Se le conoce como 7 segmentos por que cuenta con siete diodos led principales y uno extra para representar un punto. También cuenta con una carcasa para cubrirlos y 10 terminales: 2 son de alimentación (2 de Vcd o 2 de Gnd), 1 es para visualizar un punto y 7 son para representar cada uno de los números segúrn la combinación que se le ponga, estos están representados por una letra del abecedario desde la “A” hasta la letra “G”.



Funcionamiento de un display

Para poder representar los números o caracteres con este dispositivo solo basta con saber la configuración de cada una de sus leds y combinarlos.

5.DESARROLLO Y SIMULACIÓN

Enunciado 1:

Las normas de seguridad de los modernos aviones exigen que, para señales de vital importancia para la seguridad del aparato, los circuitos deben estar triplicados para que el fallo de uno de ellos no produzca una catástrofe. En caso de que los tres circuitos no produzcan la misma salida, ésta se escogerá mediante votación. Diseñe el circuito "votador" que ha de utilizarse para obtener como resultado el valor mayoritario de las tres entradas.

Análisis:

El proceso de votación consiste en tomar el valor mayoritario de las entradas.

De esta forma, la salida, f, del circuito tendrá la siguiente codificación :

- f = 0 si hay más ceros que unos en las entradas
- f = 1 si hay más unos que ceros en las entradas

El circuito votador tiene tres señales de entrada : a,b y c, que son las salidas de los circuitos triplicados .

	A	B	C	Z	Y	X	W	U	V	
0	0	0	0	0	0	0	0	0	0	0
1	0	0	1	0	0	0	0	0	1	1
2	0	1	0	0	0	0	1	0	0	4
3	0	1	1	0	0	0	1	0	1	9
4	1	0	0	0	1	1	0	0	0	16
5	1	0	1	0	1	1	0	0	1	25
6	1	1	0	1	0	0	1	0	0	36
7	1	1	1	1	1	0	0	0	1	49

$$Z = ABC' + ABC \Rightarrow AB(C + C')$$

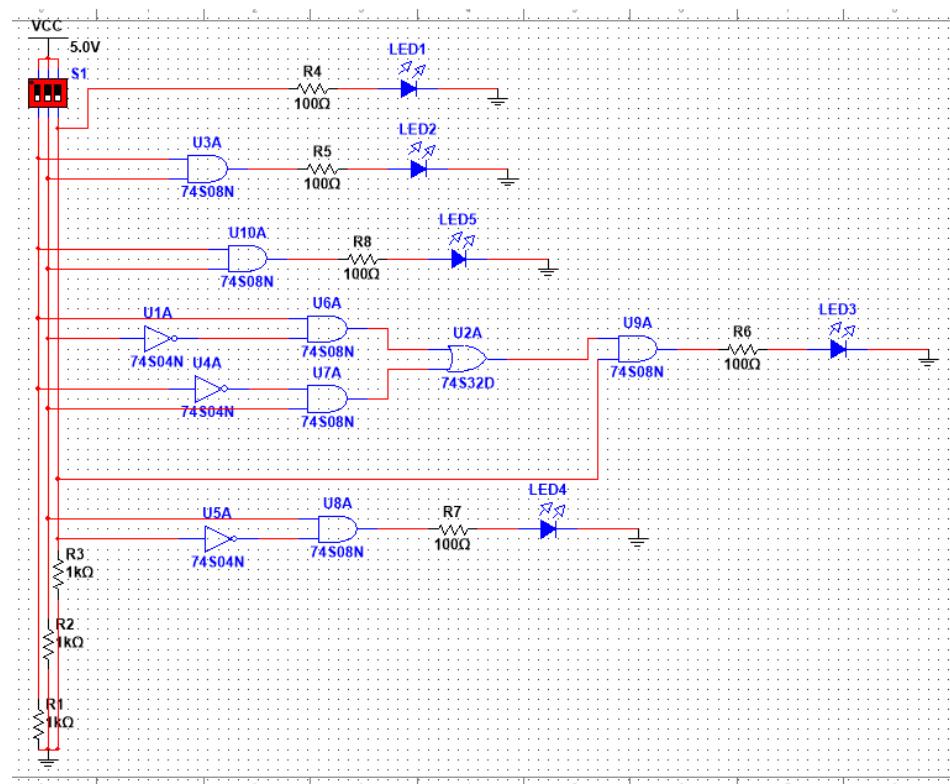
$$Y = AB'C' + AB'C \Rightarrow AB'(C' + C)$$

$$X = A'BC + AB'C \Rightarrow C(A'B + AB')$$

$$W = A'BC? + ABC? \Rightarrow A' + A) BC'$$

$$U = 0 \Rightarrow 0$$

$$V = A'B'C + A'BC + AB'C + ABC = C$$



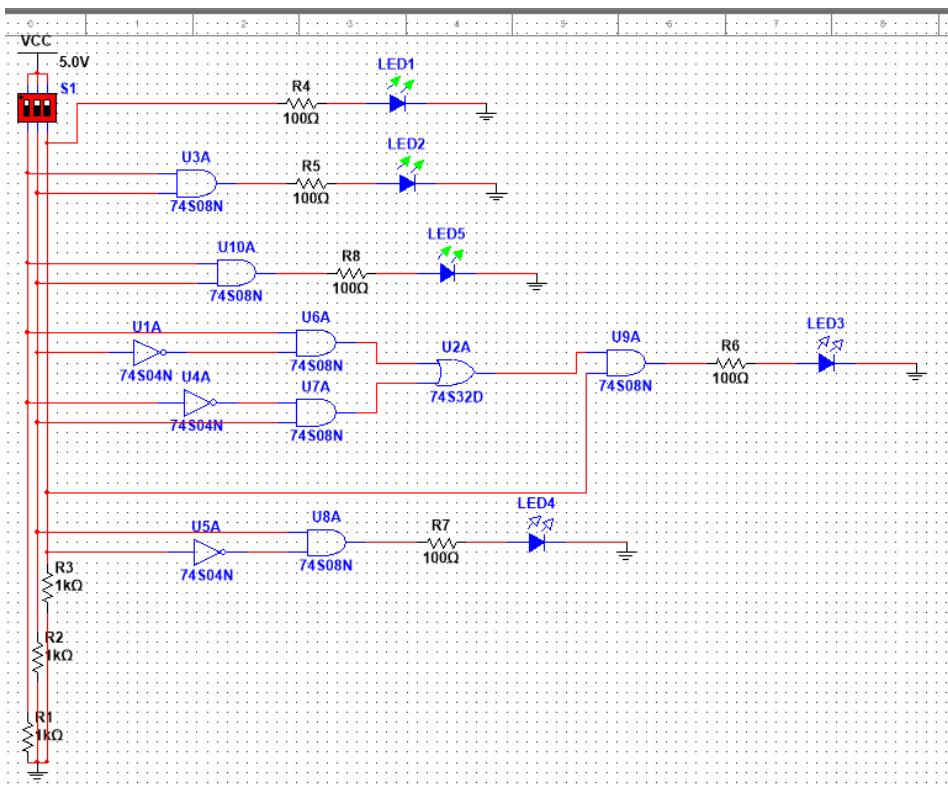
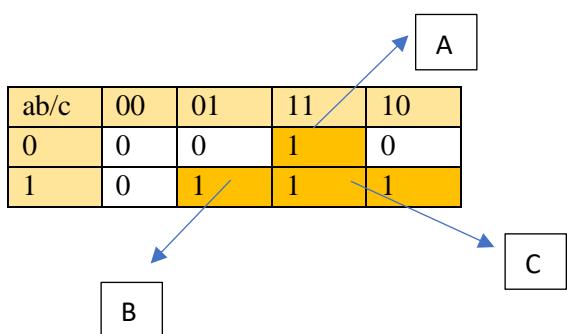


Tabla de verdad:

a	b	c	f
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

Simplificación mediante el mapa de Karnaugh



	\bar{c}	c
--	-----------	---

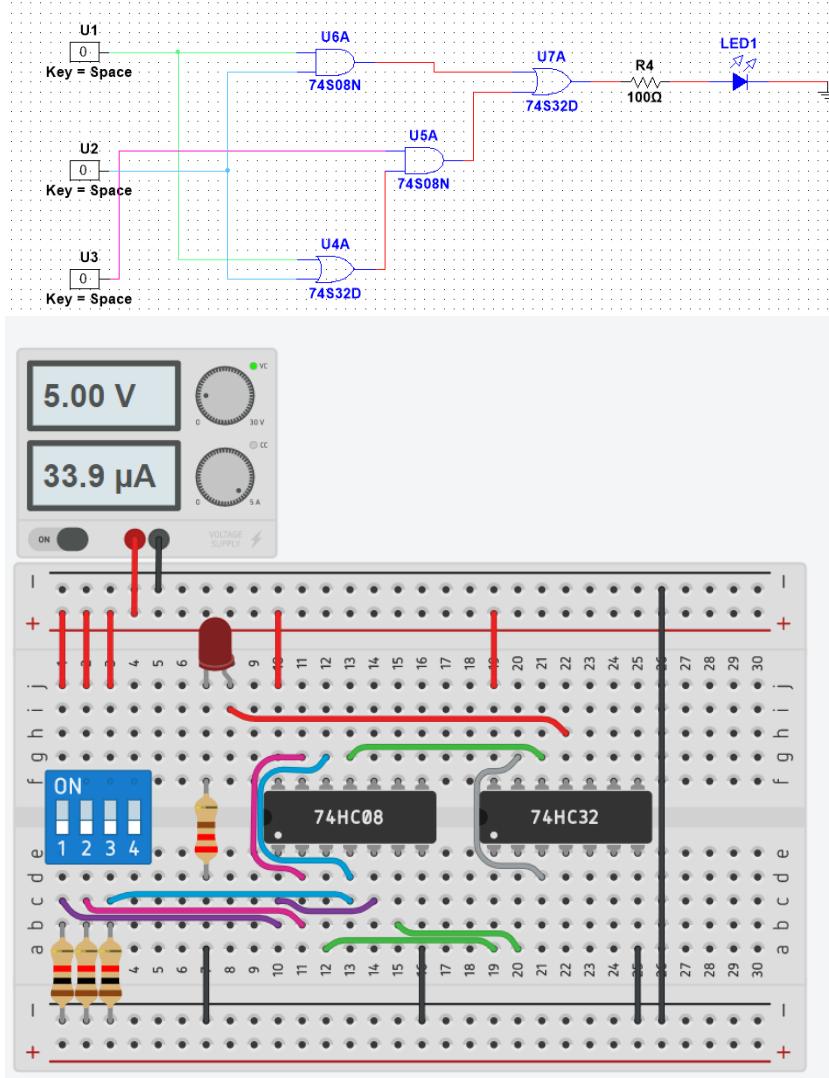
$\bar{a}\bar{b}$		
$\bar{a}b$		1
ab	1	1
$a\bar{b}$		1

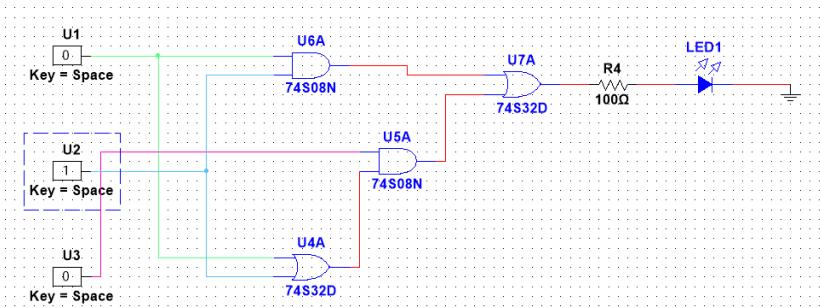
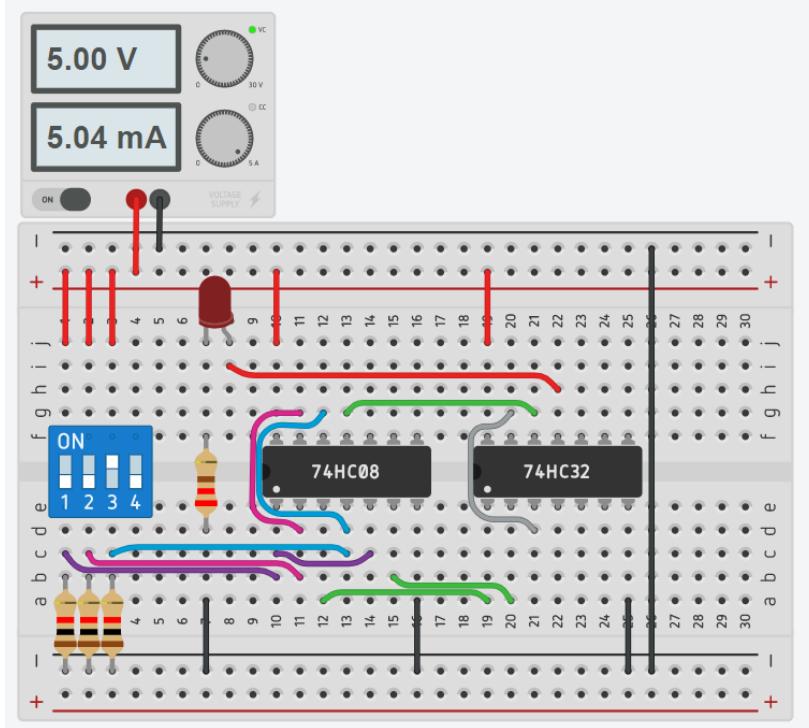
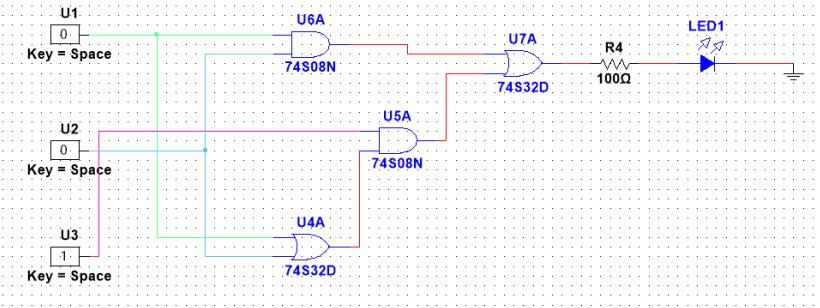
$$f = A + B + C = ab + bc + ac$$

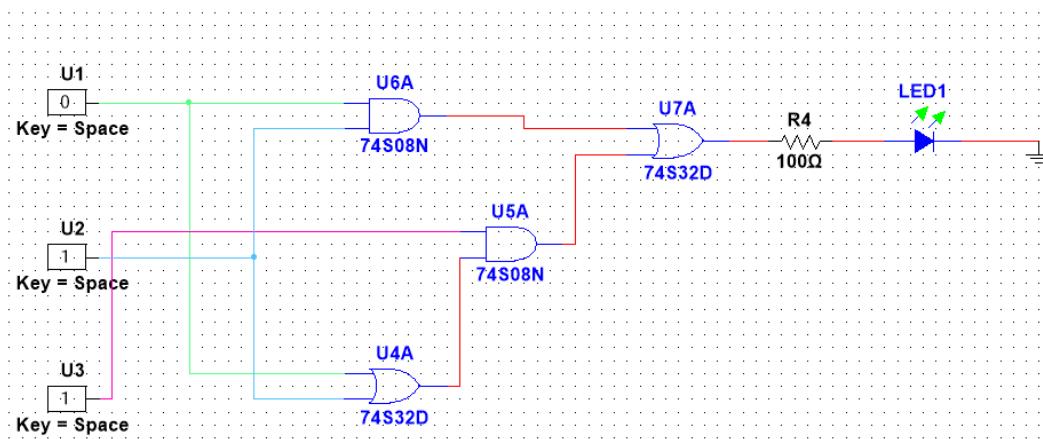
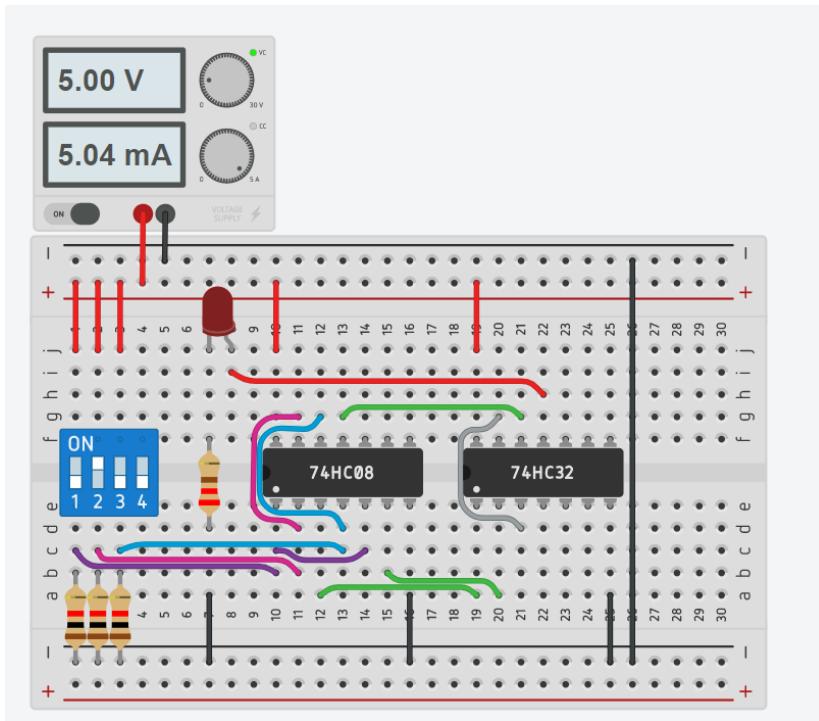
$$f = a \cdot b + c \cdot (a + b)$$

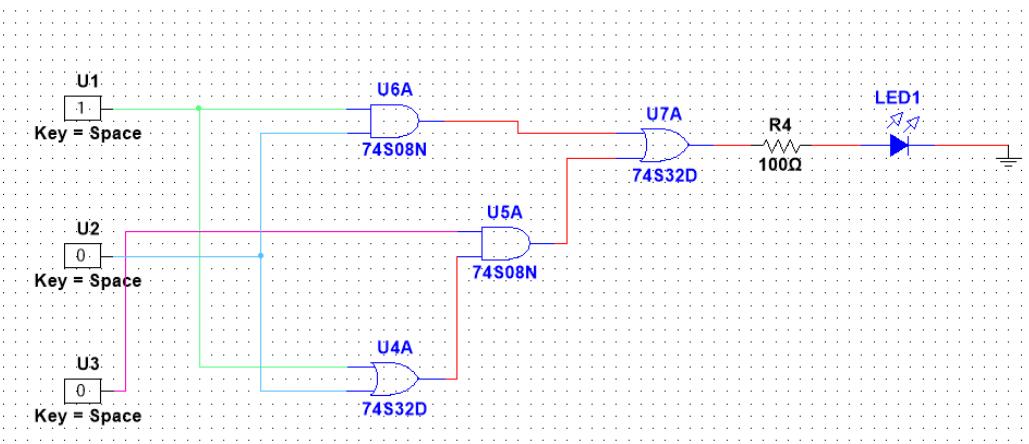
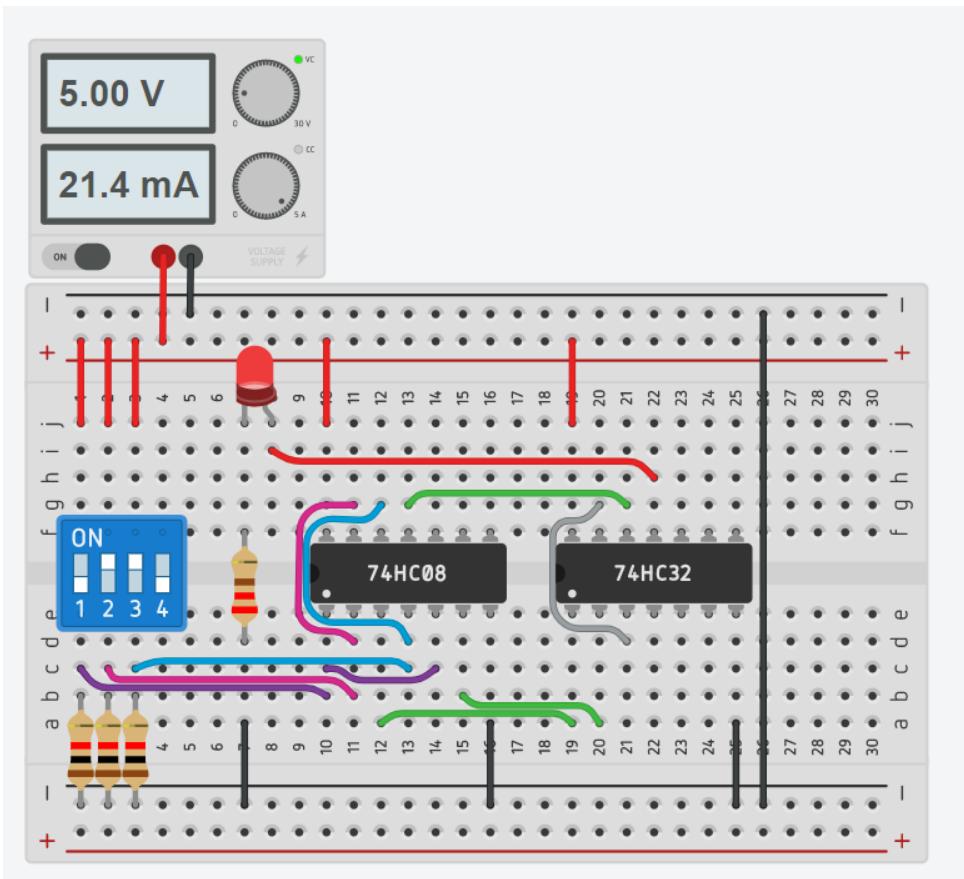
$$f = ab + ac + bc$$

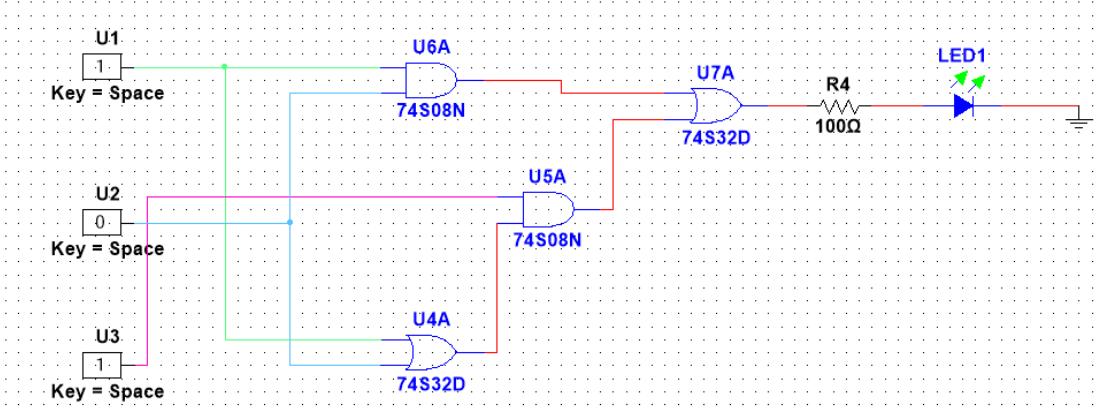
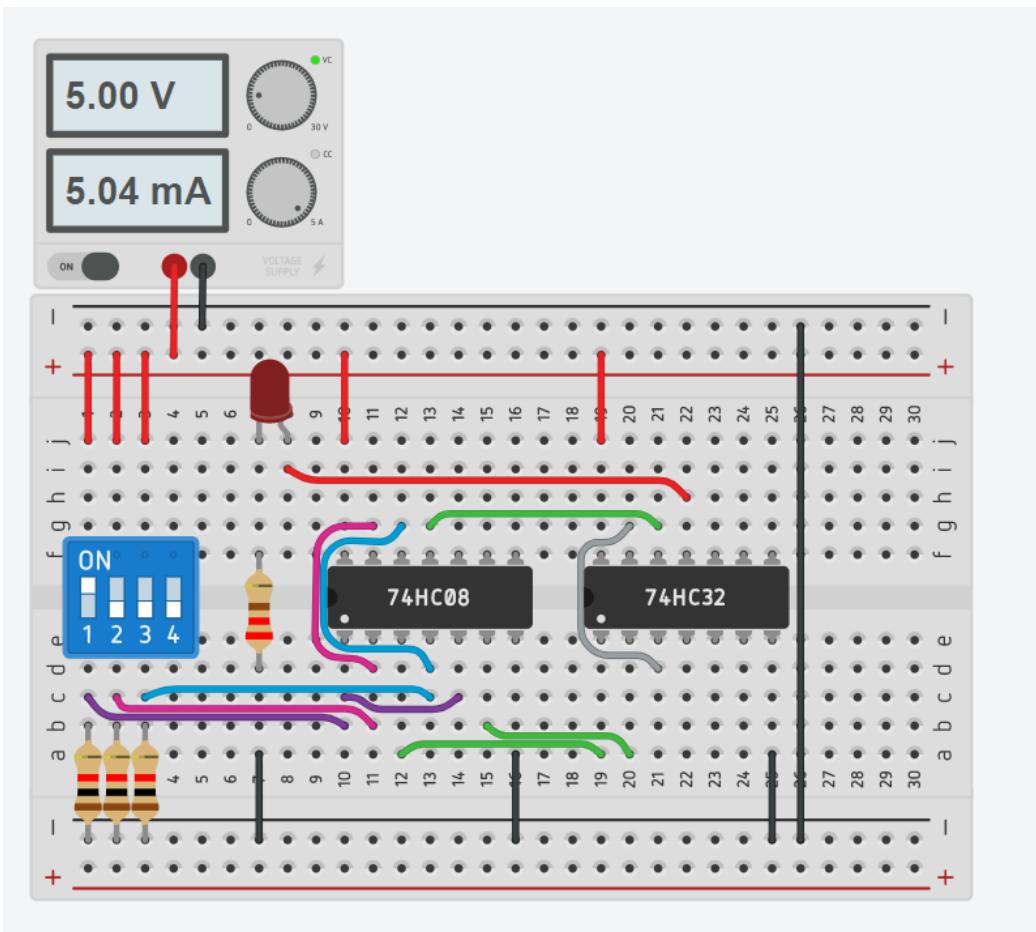
Simulación y Diagramas

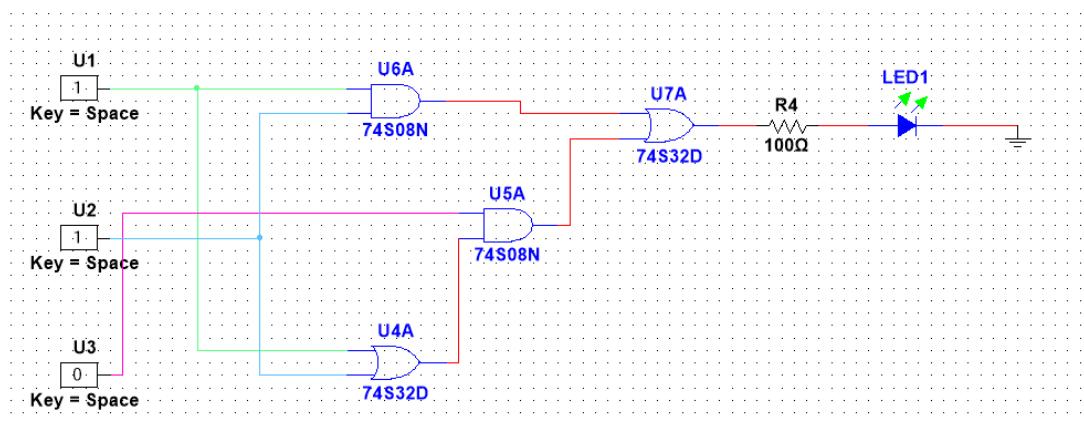
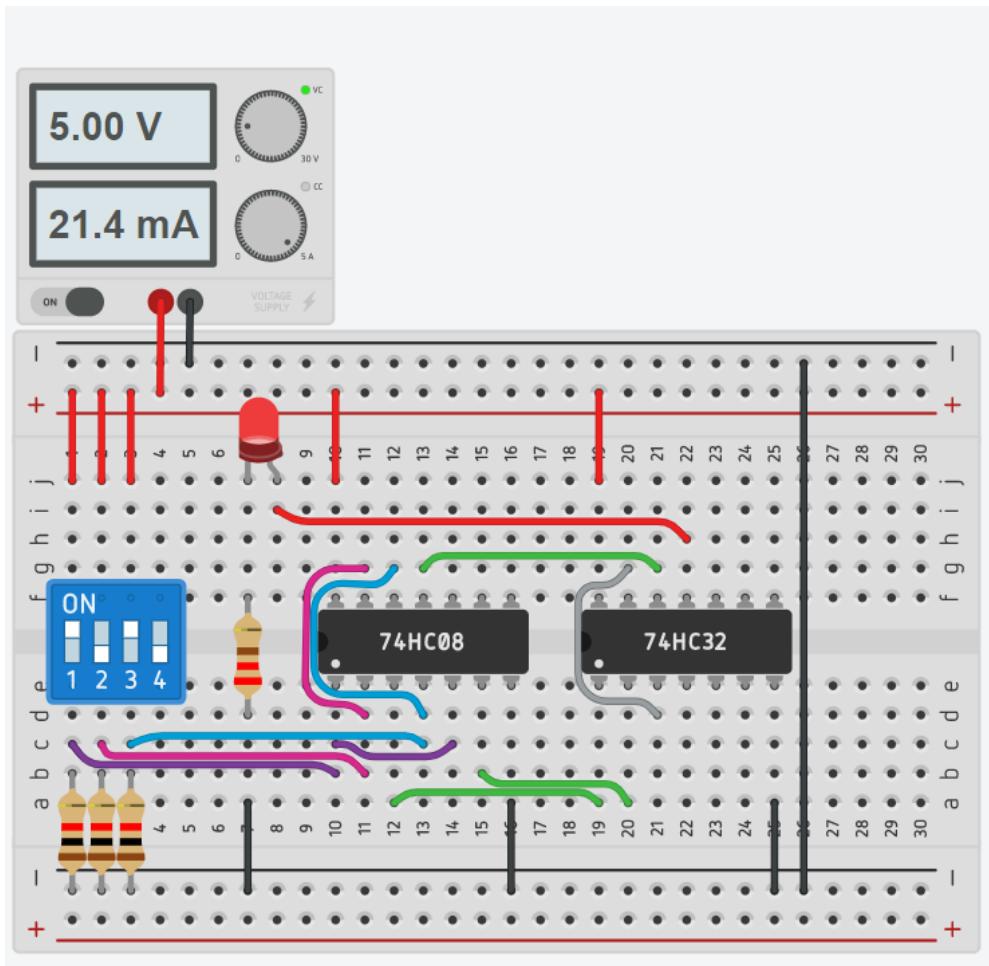


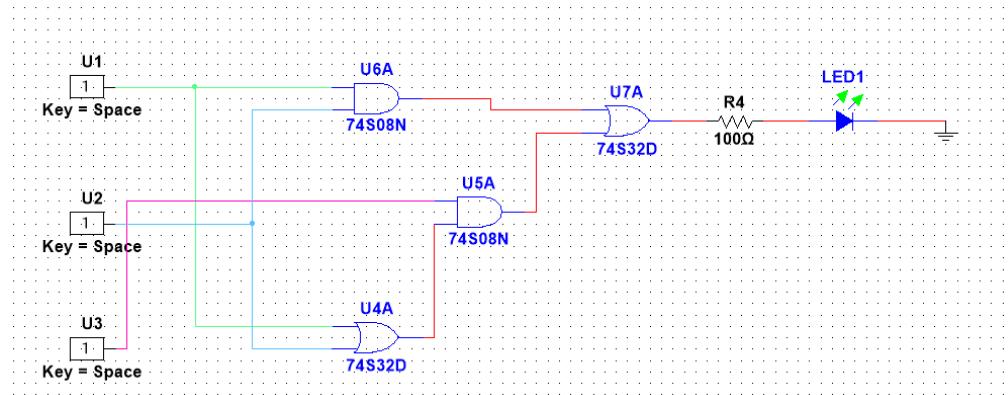
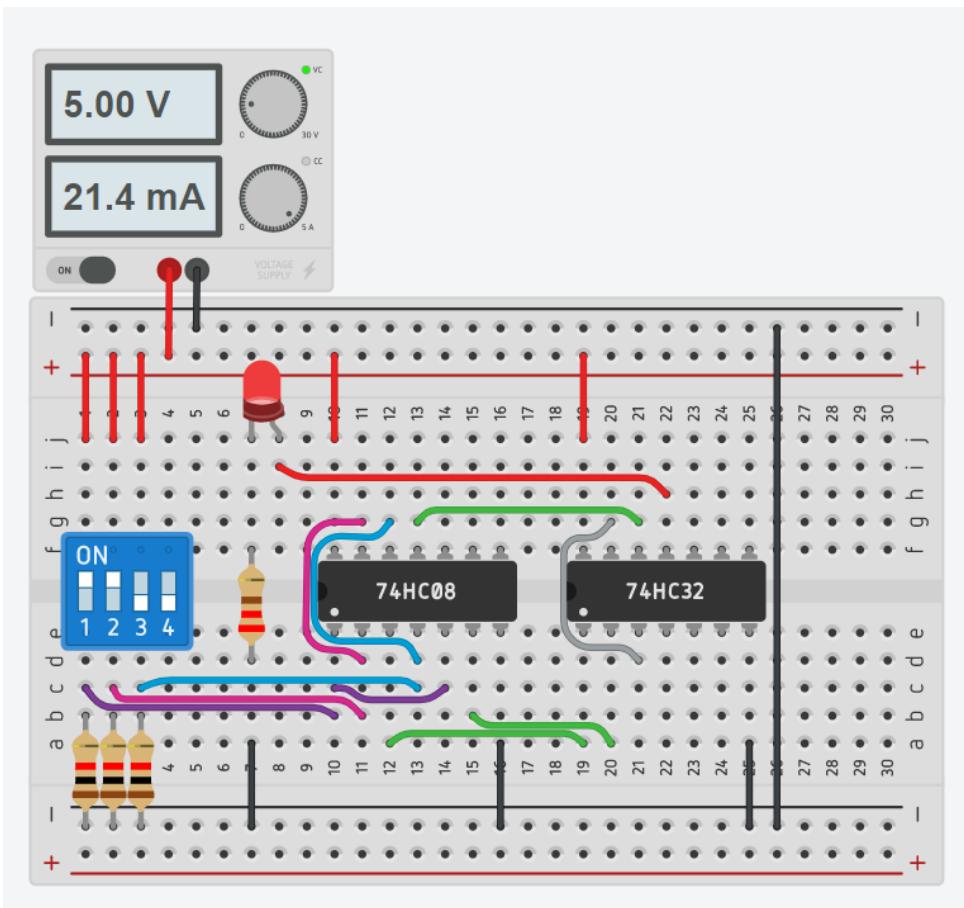


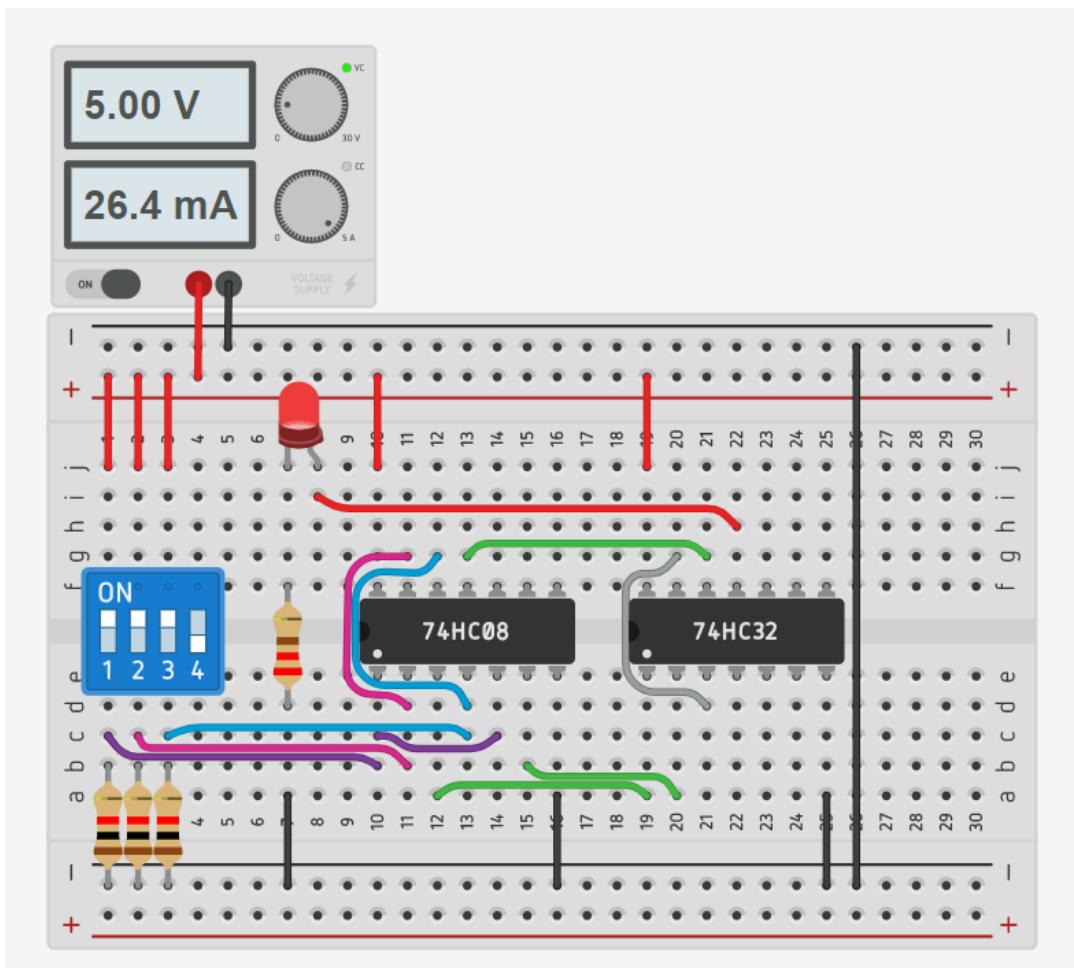












Enunciado 2:

El horario laboral de una factoría es de 8 horas diarias, divididas en tres turnos: de 8 a 11 (primer turno), de 11 a 13 (segundo turno), de 13 a 16 (descanso) y de 16 a 19 (tercer turno). Se pretende diseñar un circuito que tenga como entradas la representación binaria de la hora actual menos ocho y que proporcione a la salida el número de turno que está trabajando (si procede) o "0" si es hora de descanso.

Diseñe un módulo adicional que permita visualizar en un display de 7 segmentos el número del turno que corresponda.

Análisis

Para implementar este circuito consideramos que cada turno se desarrolla en un lapso de tiempo específico considerando la hora laboral como: 8 a 9, 9 a 10, 10 a 11,18 a 19 quedando un total de 11 horas posibles.

En un primer análisis la HORA ACTUAL son números decimales del 8 al 18 que al momento de cumplir la condición impuesta por el enunciado ($HORA ACTUAL - 8$), queda un intervalo del 0 al 10, finalmente las ENTRADAS que se van a utilizar para el desarrollo del problema es la representación en binario del último intervalo (0 al 10).

Ejemplo:

Hora actual: 8 Hora actual – 8: $8 - 8 = 0$ Representación binaria: 0 0 0 0

Hora actual: 9 Hora actual – 8: $9 - 8 = 1$ Representación binaria: 0 0 0 1

Así sucesivamente hasta completar el intervalo, para la representación binaria del 10 se necesita una entrada de 4 bits.

TURNO	HORAS	Hora actual - 8	BINARIOS ENTRADAS				Número de turno		SALIDAS DISPLAY						
			A3	A2	A1	A0	salida 2	salida 1	a	b	c	d	e	f	g
1er TURNO	8	0	0	0	0	0	0	1	0	1	1	0	0	0	0
	9	1	0	0	0	1	0	1	0	1	1	0	0	0	0
	10	2	0	0	1	0	0	1	0	1	1	0	0	0	0
2do TURNO	11	3	0	0	1	1	1	0	1	1	0	1	1	0	1
	12	4	0	1	0	0	1	0	1	1	0	1	1	0	1
DESCANSO	13	5	0	1	0	1	0	0	1	1	1	1	1	1	0
	14	6	0	1	1	0	0	0	1	1	1	1	1	1	0
	15	7	0	1	1	1	0	0	1	1	1	1	1	1	0
3er TURNO	16	8	1	0	0	0	1	1	1	1	1	1	0	0	1
	17	9	1	0	0	1	1	1	1	1	1	1	0	0	1
	18	10	1	0	1	0	1	1	1	1	1	1	0	0	1

Al desarrollar la tabla de verdad la combinación de la salida 2 y salida 1 arrojan en binario lo siguiente ("1" encendido y "0" apagado):

Primer turno: 01

Segundo turno: 10

Tercer turno: 11

Descanso: 00

- **Para salida 2**

min-Términos (3, 4, 8, 9, 10)

$$\text{salida 2} = F(a_0, a_1, a_2, a_3)$$

$$\text{salida 2} =$$

$$= \overline{a_3} \cdot \overline{a_2} \cdot a_1 \cdot a_0 + \overline{a_3} \cdot a_2 \cdot \overline{a_1} \cdot \overline{a_0} + a_3 \cdot \overline{a_2} \cdot \overline{a_1} \cdot \overline{a_0} + a_3 \cdot \overline{a_2} \cdot \overline{a_1} \cdot a_0 + a_3 \cdot \overline{a_2} \cdot a_1 \cdot \overline{a_0}$$

$$= \overline{a_3} \cdot \overline{a_2} \cdot a_1 \cdot a_0 + \overline{a_3} \cdot a_2 \cdot \overline{a_1} \cdot \overline{a_0} + a_3 \cdot \overline{a_2} \cdot \overline{a_1} \cdot (\overline{a_0} + a_0) + a_3 \cdot \overline{a_2} \cdot a_1 \cdot \overline{a_0}$$

$$= \overline{a_3} \cdot \overline{a_2} \cdot a_1 \cdot a_0 + \overline{a_3} \cdot a_2 \cdot \overline{a_1} \cdot \overline{a_0} + a_3 \cdot \overline{a_2} \cdot \overline{a_1} + a_3 \cdot \overline{a_2} \cdot a_1 \cdot \overline{a_0}$$

$$= a_3 \cdot (\overline{a_2} \cdot \overline{a_1} + \overline{a_2} \cdot a_1 \cdot \overline{a_0}) + \overline{a_3} \cdot \overline{a_2} \cdot a_1 \cdot a_0 + \overline{a_3} \cdot a_2 \cdot \overline{a_1} \cdot \overline{a_0}$$

$$= a_3 \cdot [\overline{a_2} \cdot (\overline{a_1} + a_1 \cdot \overline{a_0})] + \overline{a_3} \cdot \overline{a_2} \cdot a_1 \cdot a_0 + \overline{a_3} \cdot a_2 \cdot \overline{a_1} \cdot \overline{a_0}$$

$$= a_3 \cdot [\overline{a_2} \cdot ((\overline{a_1} + a_1) \cdot (\overline{a_1} \cdot \overline{a_0}))] + \overline{a_3} \cdot \overline{a_2} \cdot a_1 \cdot a_0 + \overline{a_3} \cdot a_2 \cdot \overline{a_1} \cdot \overline{a_0}$$

$$= a_3 \cdot (\overline{a_2} \cdot \overline{a_1} + \overline{a_2} \cdot \overline{a_0}) + \overline{a_3} \cdot \overline{a_2} \cdot a_1 \cdot a_0 + \overline{a_3} \cdot a_2 \cdot \overline{a_1} \cdot \overline{a_0}$$

$$a_3 \cdot \overline{a_2} \cdot \overline{a_1} + a_3 \cdot \overline{a_2} \cdot \overline{a_0} + \overline{a_3} \cdot \overline{a_2} \cdot a_1 \cdot a_0 + \overline{a_3} \cdot a_2 \cdot \overline{a_1} \cdot \overline{a_0}$$

- Para salida 1

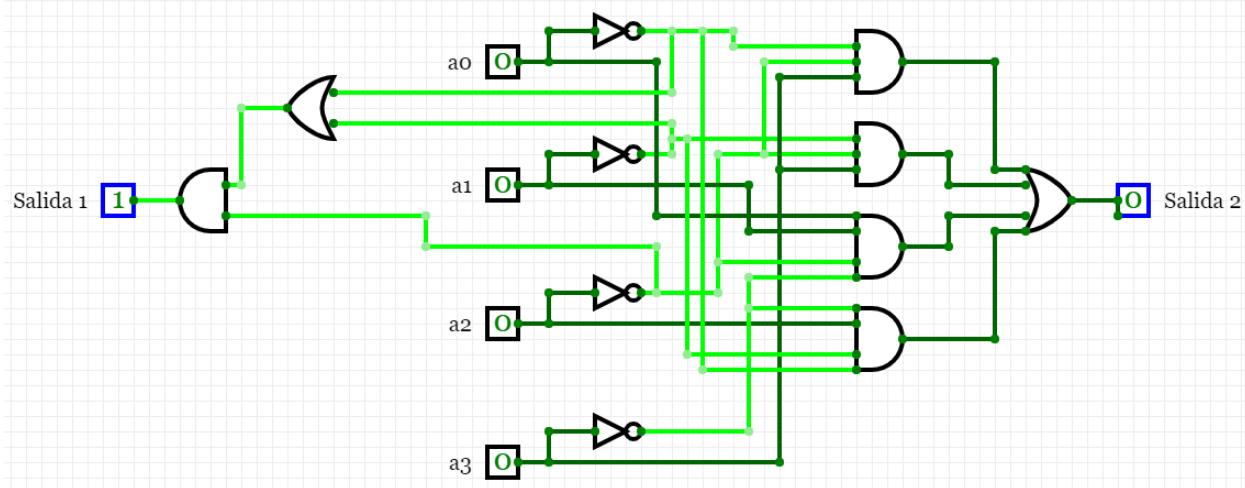
min-Términos (0, 1, 2, 8, 9, 10)

$$\text{salida 1} = F(a_0, a_1, a_2, a_3)$$

salida 1 =

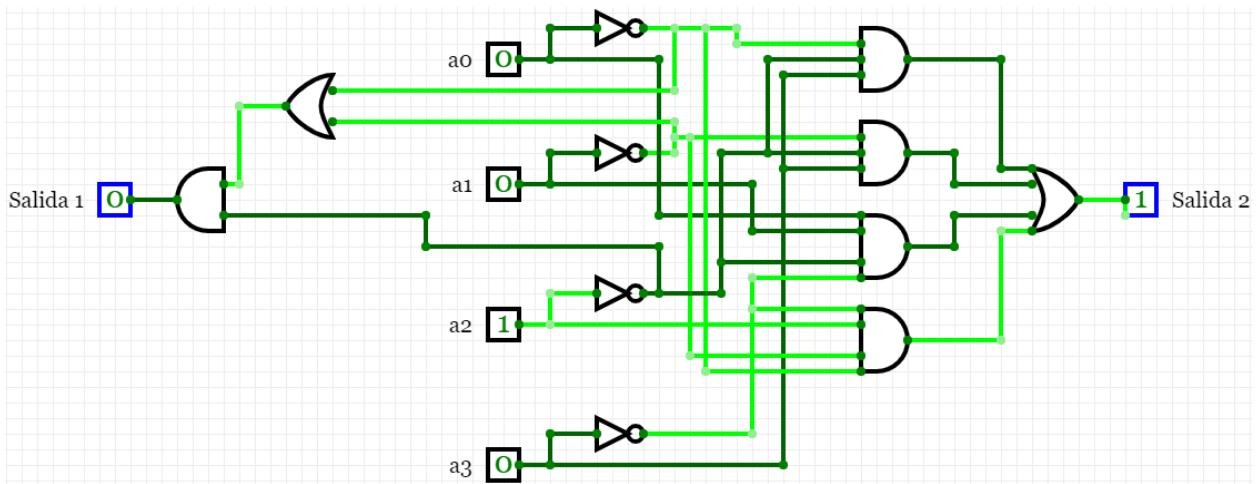
$$\begin{aligned}
 &= \overline{a_3} \cdot \overline{a_2} \cdot \overline{a_1} \cdot \overline{a_0} + \overline{a_3} \cdot \overline{a_2} \cdot \overline{a_1} \cdot a_0 + \overline{a_3} \cdot \overline{a_2} \cdot a_1 \cdot \overline{a_0} + a_3 \cdot \overline{a_2} \cdot \overline{a_1} \cdot \overline{a_0} + a_3 \cdot \overline{a_2} \cdot \overline{a_1} \cdot a_0 + a_3 \cdot \overline{a_2} \cdot a_1 \cdot \overline{a_0} \\
 &= \overline{a_3} \cdot \overline{a_2} \cdot \overline{a_1} \cdot (\overline{a_0} + a_0) + \overline{a_2} \cdot a_1 \cdot \overline{a_0} \cdot (\overline{a_3} + a_3) + a_3 \cdot \overline{a_2} \cdot \overline{a_1} \cdot (\overline{a_0} + a_0) \\
 &= \overline{a_3} \cdot \overline{a_2} \cdot \overline{a_1} + \overline{a_2} \cdot a_1 \cdot \overline{a_0} + a_3 \cdot \overline{a_2} \cdot \overline{a_1} \\
 &= \overline{a_2} \cdot \overline{a_1} \cdot (\overline{a_3} + a_3) + \overline{a_2} \cdot a_1 \cdot \overline{a_0} \\
 &= \overline{a_2} \cdot \overline{a_1} + \overline{a_2} \cdot a_1 \cdot \overline{a_0} \\
 &= \overline{a_2} \cdot (\overline{a_1} + (a_1 \cdot \overline{a_0})) \\
 &= \overline{a_2} \cdot ((\overline{a_1} + a_1) \cdot (\overline{a_1} + \overline{a_0})) \\
 &= \overline{a_2} \cdot (\overline{a_1} + \overline{a_0})
 \end{aligned}$$

Simulación y Diagramas

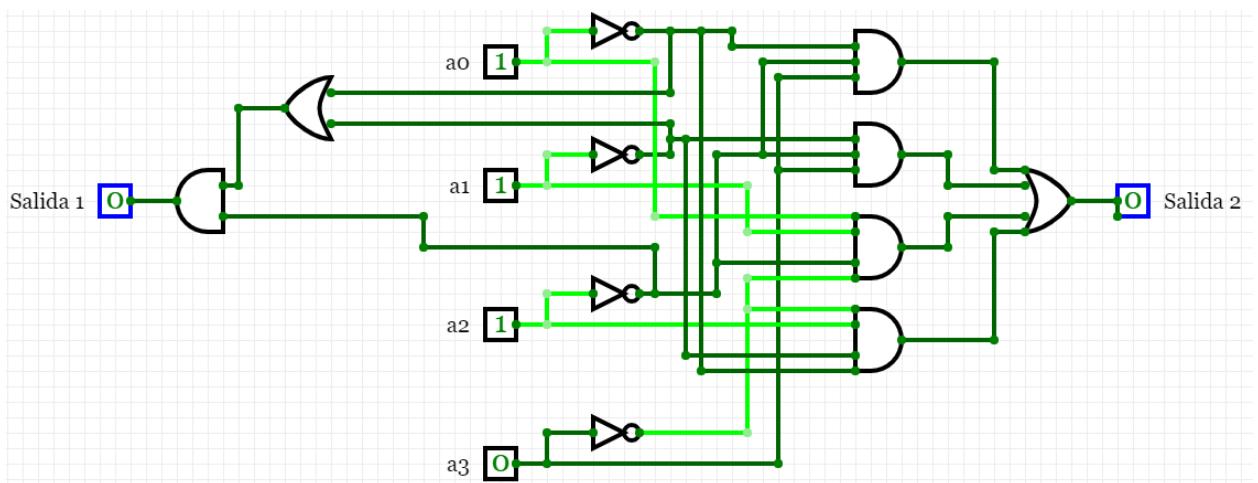


Primer turno

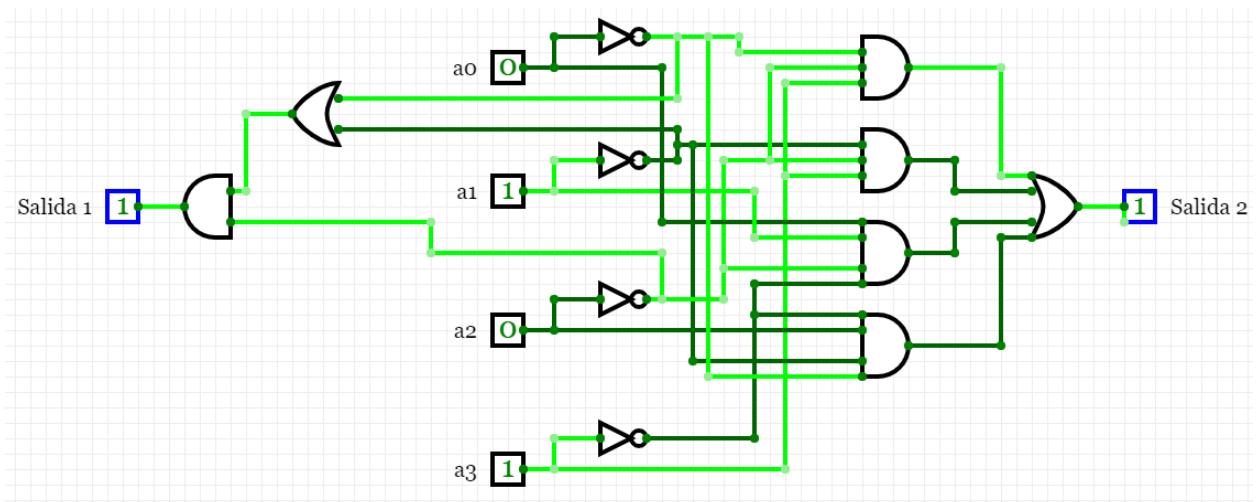
Segundo turno



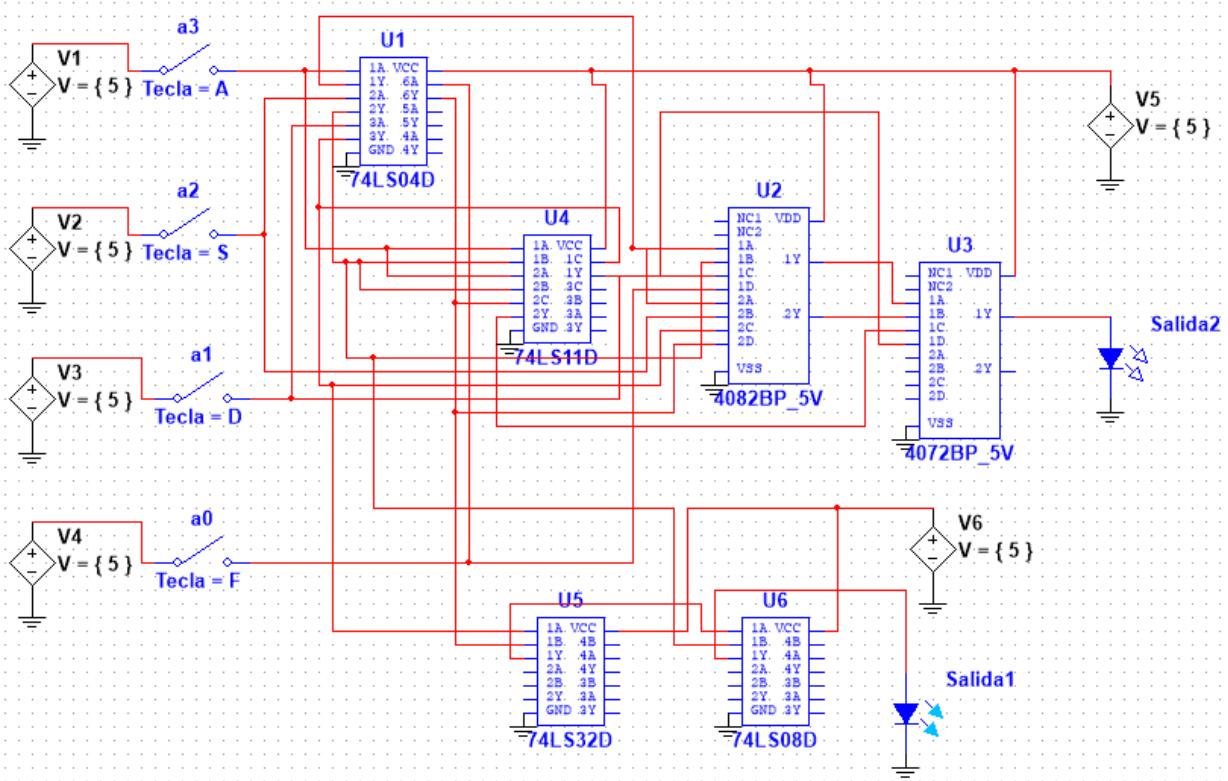
Descanso



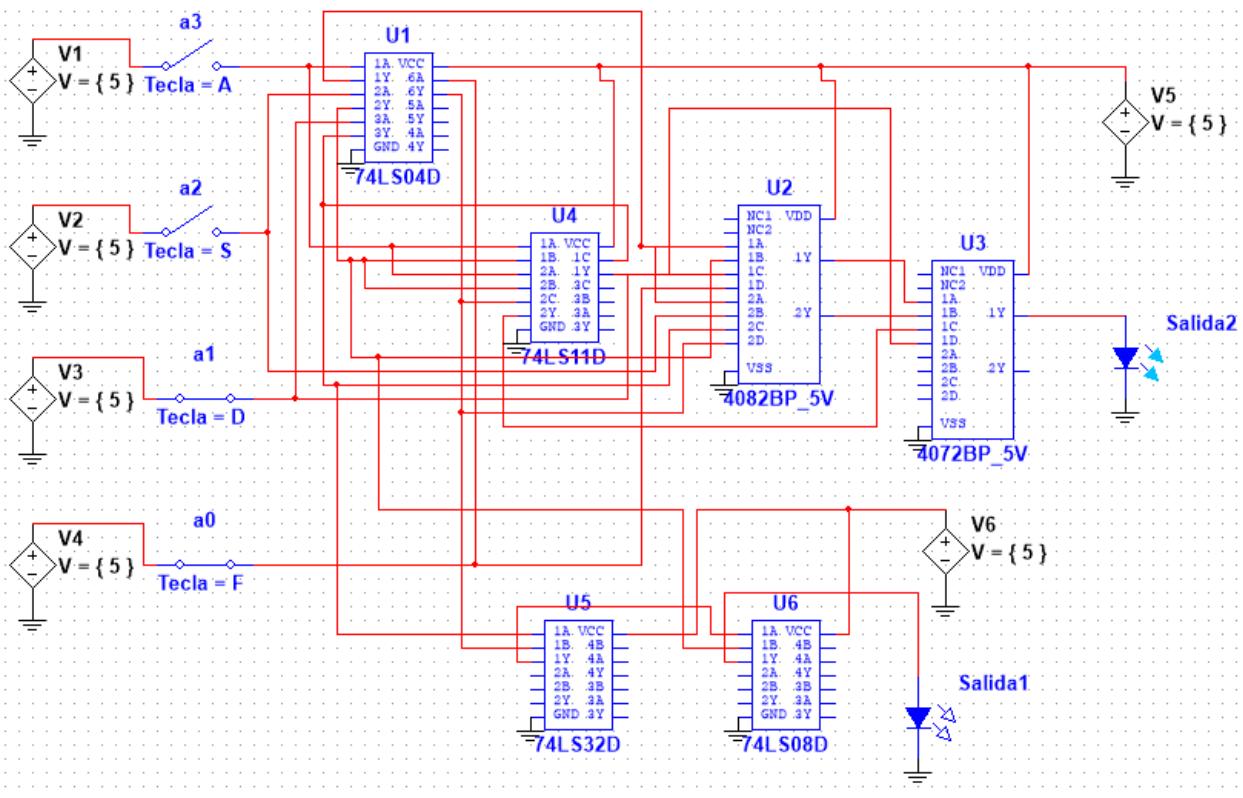
Tercer turno



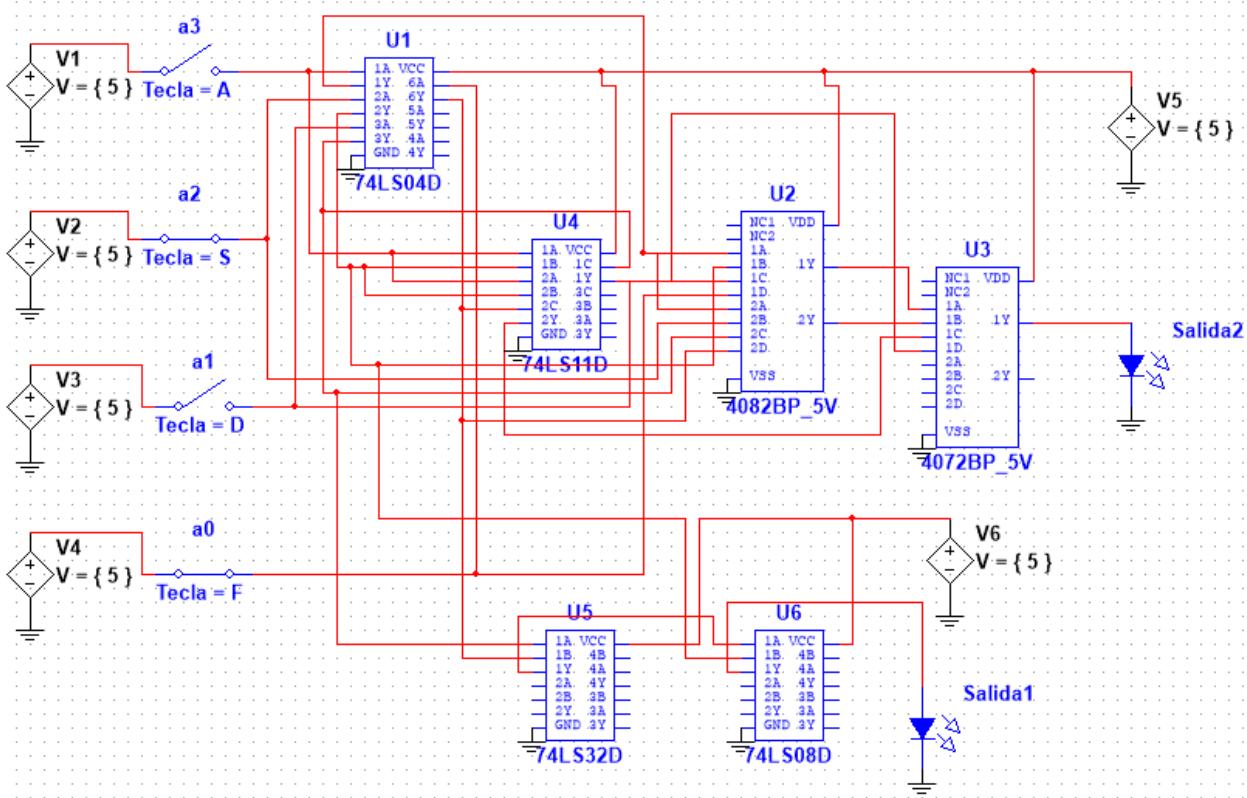
Primer turno



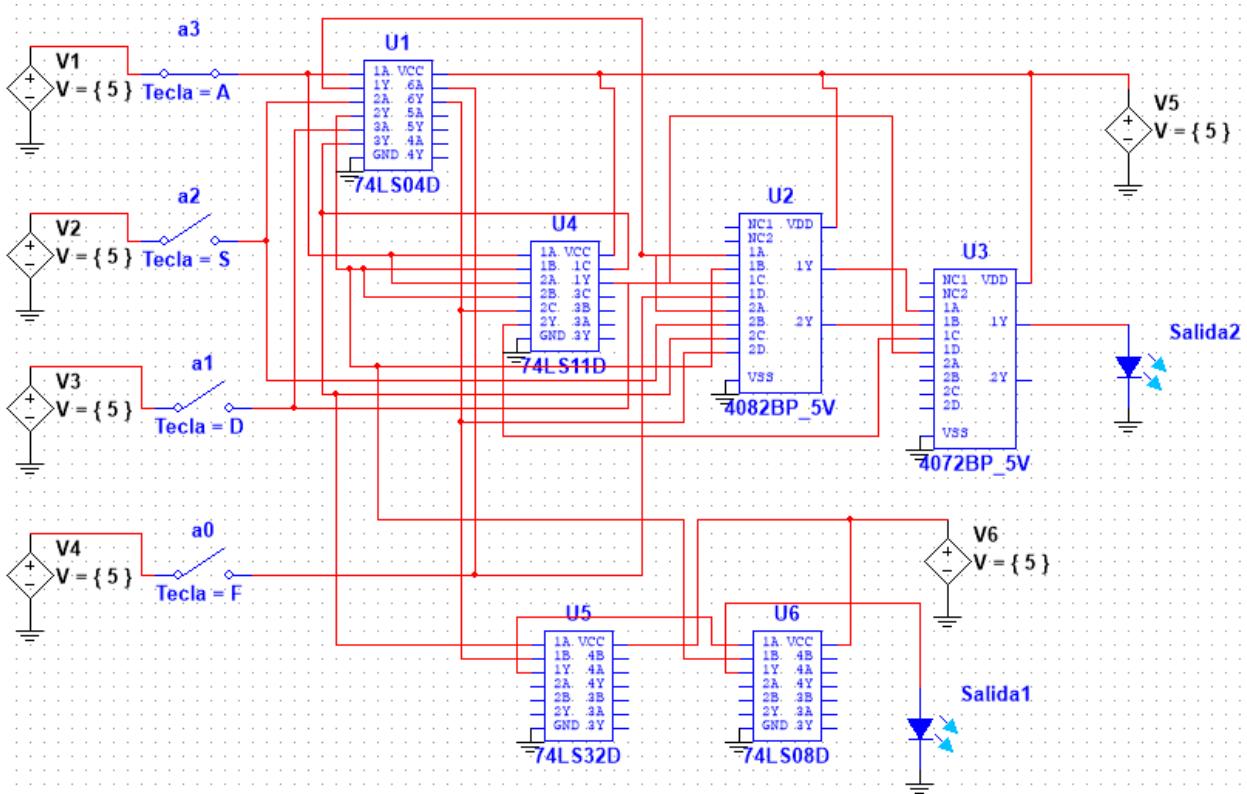
Segundo turno



Descanso



Tercer turno



Enunciado 3:

Se
diseñar un

pretende
circuito

Entradas		B		Salidas		
A		b1	b0	A>B	A=B	A<B
a1	a0			M	L	m
0	0	0	0	0	1	0
0	0	0	1	0	0	1
0	0	1	0	0	0	1
0	0	1	1	0	0	1
0	1	0	0	1	0	0
0	1	0	1	0	1	0
0	1	1	0	0	0	1
1	0	0	0	1	0	0
1	0	0	1	1	0	0
1	0	1	0	0	1	0
1	0	1	1	0	0	1
1	1	0	0	1	0	0
1	1	0	1	1	0	0
1	1	1	0	1	0	0
1	1	1	1	0	1	0

comparador de 2 números de 2 bits, A = (a1, a0) y B = (b1, b0). Dicho circuito deberá tener tres salidas M, L, m, de tal forma que:

$$M = 1 \text{ si } A > B$$

$$L = 1 \text{ si } A = B$$

$$m = 1 \text{ si } A < B$$

Diseñe exclusivamente con puertas NOR.

Análisis:

Funciones resultantes aplicando min términos.

- $M = 0100 + 1000 + 1001 + 1100 + 1101 + 1110$

$$M = \overline{a_1} a_0 \overline{b_1} \overline{b_0} + a_1 \overline{a_0} \overline{b_1} \overline{b_0} + a_1 \overline{a_0} \overline{b_1} b_0 + a_1 a_0 \overline{b_1} \overline{b_0} + a_1 a_0 \overline{b_1} b_0 + a_1 a_0 b_1 \overline{b_0}$$

- $L = 0000 + 0101 + 1010 + 1111$

$$L = \overline{a_1} \overline{a_0} \overline{b_1} \overline{b_0} + \overline{a_1} a_0 \overline{b_1} b_0 + a_1 \overline{a_0} b_1 \overline{b_0} + a_1 a_0 b_1 b_0$$

- $m = 0001 + 0010 + 0011 + 0110 + 0111 + 1011$

$$m = \overline{a_1} \overline{a_0} \overline{b_1} b_0 + \overline{a_1} \overline{a_0} b_1 \overline{b_0} + \overline{a_1} \overline{a_0} b_1 b_0 + \overline{a_1} a_0 b_1 \overline{b_0} + \overline{a_1} a_0 b_1 b_0 + a_1 \overline{a_0} b_1 b_0$$

Simplificación de términos mediante el Mapas de Karnaugh

$$M = \overline{a_1} a_0 \overline{b_1} \overline{b_0} + a_1 \overline{a_0} \overline{b_1} \overline{b_0} + a_1 \overline{a_0} \overline{b_1} b_0 + a_1 a_0 \overline{b_1} \overline{b_0} + a_1 a_0 \overline{b_1} b_0 + a_1 a_0 b_1 \overline{b_0}$$

$\frac{b_1 b_0}{a_1 a_0}$	00	01	11	10
00				
01	1			
11	1	1		1
10	1	1		

$$M = a_0 \bar{b_1} \bar{b_0} + a_1 a_0 \bar{b_0} + a_1 \bar{b_1}$$

$$L = \bar{a_1} \bar{a_0} \bar{b_1} \bar{b_0} + \bar{a_1} a_0 \bar{b_1} b_0 + a_1 \bar{a_0} b_1 \bar{b_0} + a_1 a_0 b_1 b_0$$

$\frac{b_1 b_0}{a_1 a_0}$	00	01	11	10
00	1			
01		1		
11			1	
10				1

$$L = \bar{a_1} \bar{a_0} \bar{b_1} \bar{b_0} + \bar{a_1} a_0 \bar{b_1} b_0 + a_1 \bar{a_0} b_1 \bar{b_0} + a_1 a_0 b_1 b_0$$

$$m = \bar{a_1} \bar{a_0} \bar{b_1} b_0 + \bar{a_1} \bar{a_0} b_1 \bar{b_0} + \bar{a_1} \bar{a_0} b_1 b_0 + \bar{a_1} a_0 b_1 \bar{b_0} + \bar{a_1} a_0 b_1 b_0 + a_1 \bar{a_0} b_1 b_0$$

$\frac{b_1 b_0}{a_1 a_0}$	00	01	11	10
00		1	1	1
01			1	1
11				
10			1	

$$m = \bar{a_1} b_1 + \bar{a_1} \bar{a_0} b_0 + \bar{a_0} b_1 b_0$$

Diseño en puertas NOR

- $M = a_0 \bar{b}_1 \bar{b}_0 + a_1 a_0 \bar{b}_0 + a_1 \bar{b}_1$

$$M = \overline{\overline{a_0} \overline{b_1} \overline{b_0}} + \overline{\overline{a_1} \overline{a_0} \overline{b_0}} + \overline{\overline{a_1} \overline{b_1}}$$

$$M = \overline{\overline{a_0} + \overline{b_1} + \overline{b_0}} + \overline{\overline{a_1} + \overline{a_0} + \overline{b_0}} + \overline{\overline{a_1} + \overline{b_1}}$$

- $L = \overline{a_1} \overline{a_0} \overline{b_1} \overline{b_0} + \overline{a_1} a_0 \overline{b_1} b_0 + a_1 \overline{a_0} b_1 \overline{b_0} + a_1 a_0 b_1 b_0$

$$L = \overline{\overline{a_1} \overline{a_0} \overline{b_1} \overline{b_0}} + \overline{\overline{a_1} a_0 \overline{b_1} b_0} + \overline{\overline{a_1} \overline{a_0} b_1 \overline{b_0}} + \overline{\overline{a_1} a_0 b_1 \overline{b_0}}$$

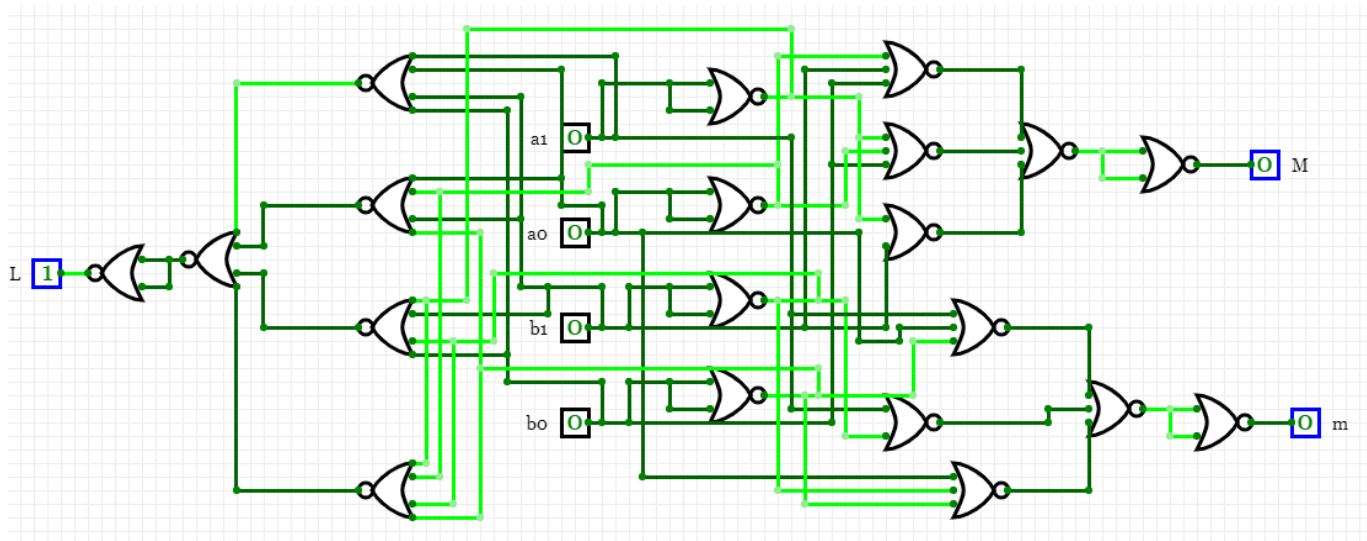
$$L = \overline{\overline{a_1} + a_0 + b_1 + b_0} + \overline{\overline{a_1} + \overline{a_0} + b_1 + \overline{b_0}} + \overline{\overline{a_1} + a_0 + \overline{b_1} + b_0} + \overline{\overline{a_1} + \overline{a_0} + \overline{b_1} + \overline{b_0}}$$

- $m = \overline{a_1} b_1 + \overline{a_1} \overline{a_0} b_0 + \overline{a_0} b_1 b_0$

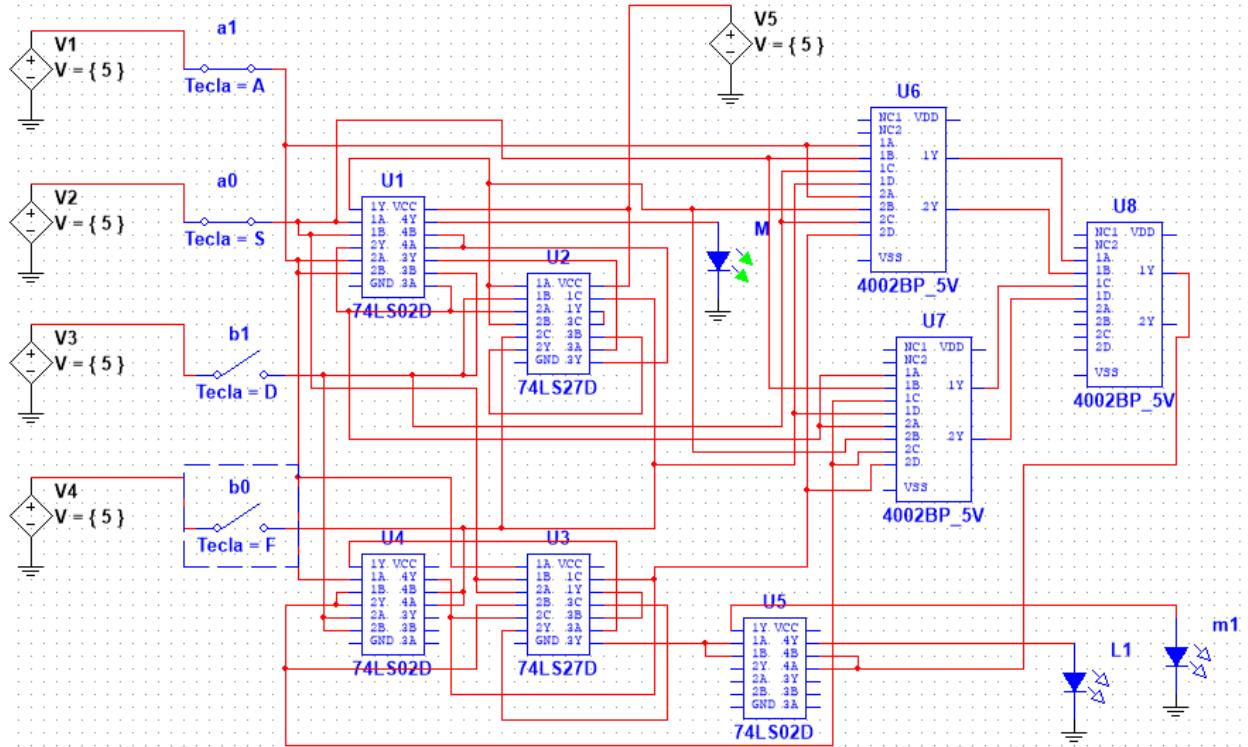
$$m = \overline{\overline{a_1} b_1} + \overline{\overline{a_1} \overline{a_0} b_0} + \overline{\overline{a_0} b_1 b_0}$$

$$m = \overline{\overline{a_1} + \overline{b_1}} + \overline{\overline{a_1} + a_0 + \overline{b_0}} + \overline{a_0 + \overline{b_1} + \overline{b_0}}$$

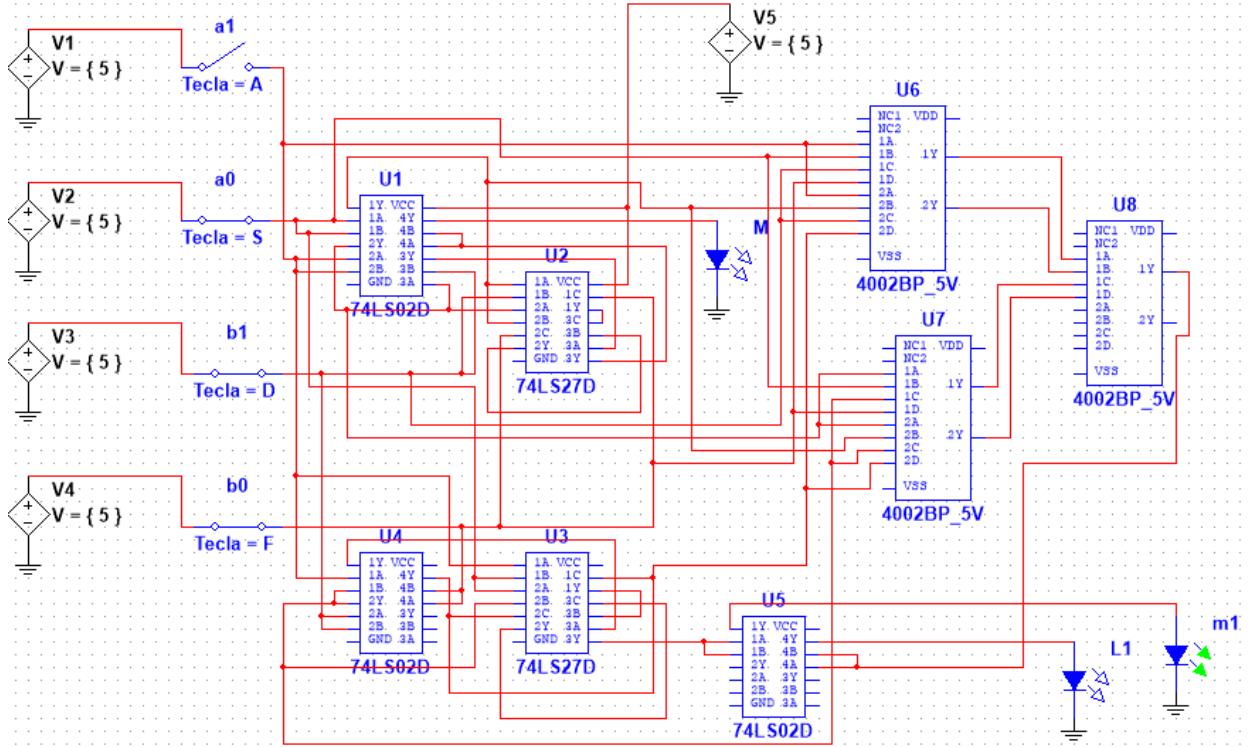
Simulación y Diagramas



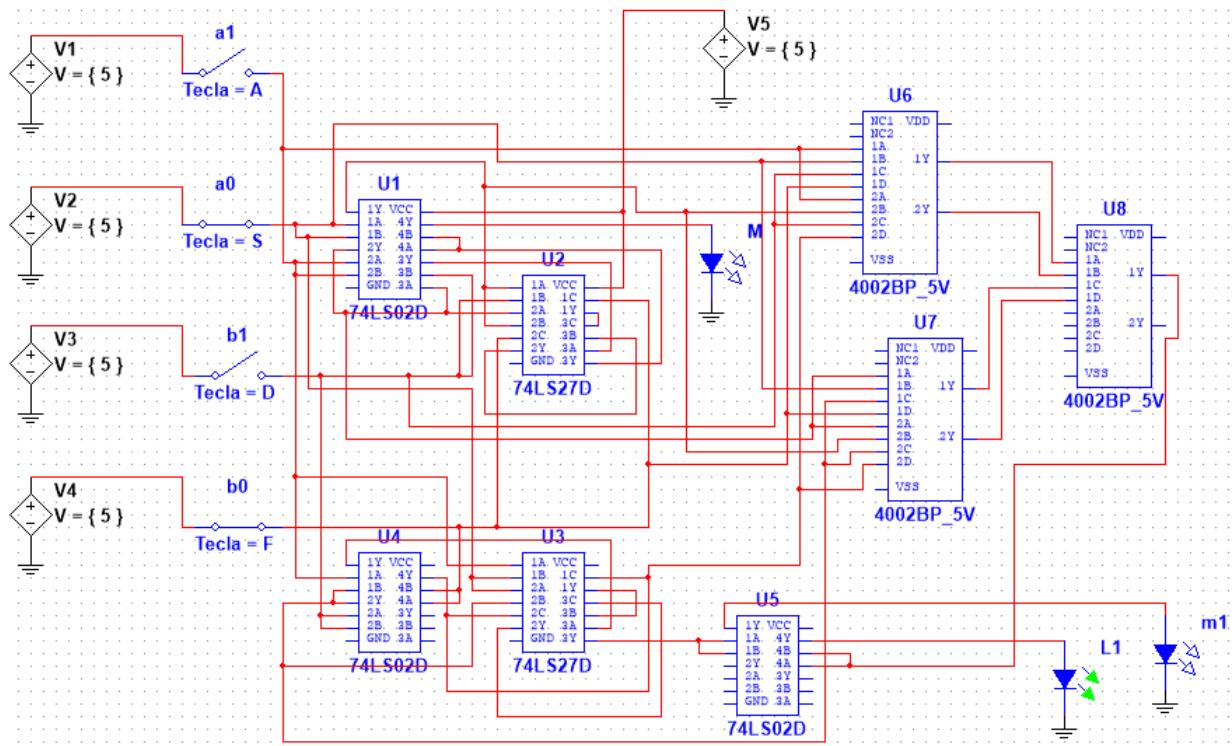
Cumple la condición de $M=1$; si $A>B$



Cumple la condición de $m=1$; si $A<B$



Cumple la condición de $L=1$; si $A=B$



7. MAPA DE VARIABLES ENFOCADO EN EL ENUNCIADO DOS

VARIABLES	FUNCIÓN OPERATIVA
<p>cuando consultar_turno .Clic ejecutar</p>	Esta variable de ejecución, nos inicializará el programa principal al momento que le demos un clic sobre el botón “CONSULTAR TURNO”
<p>poner TURNOS . Foto como</p>	Esta variable almacena la imagen del turno proyectado en el display de 7 segmentos que corresponde según sea el caso.
<p>poner entrada_4 . Texto como poner entrada_3 . Texto como poner entrada_2 . Texto como poner entrada_1 . Texto como</p>	Estas variables almacenan los números o bits ingresados por el usuario.

<pre>poner turno_en_Proceso . Texto como poner salida_horaActual . Texto como poner salida_horaActualMenos8 . Texto como poner ERROR . Texto como</pre>	<p>Las siguientes son variables de impresión, almacenan los resultados que se van a mostrar en pantalla.</p>
<pre>poner ERROR . Texto como</pre>	<p>Esta variable imprime la advertencia cuando el número ingresado por el usuario está fuera de rango.</p>
<pre>cuando nuevo_turno .Presionar ejecutar</pre>	<p>Es una variable de ejecución que inicializa o limpia todas las variables del programa para un nuevo turno.</p>
<pre>cuando salir .Clic ejecutar cerrar la aplicación</pre>	<p>Es una variable de ejecución finaliza la aplicación y la cierra.</p>

8.EXPLICACIÓN DEL CÓDIGO FUENTE ENFOCADO EN EL ENUNCIADO DOS

1. cuando consultar_turno. Clic

ejecutar

Este botón guarda una serie de instrucciones para realizar la asignación de turnos, las mismas que se ejecutarán siempre y cuando le demos un clic.

si

entonces

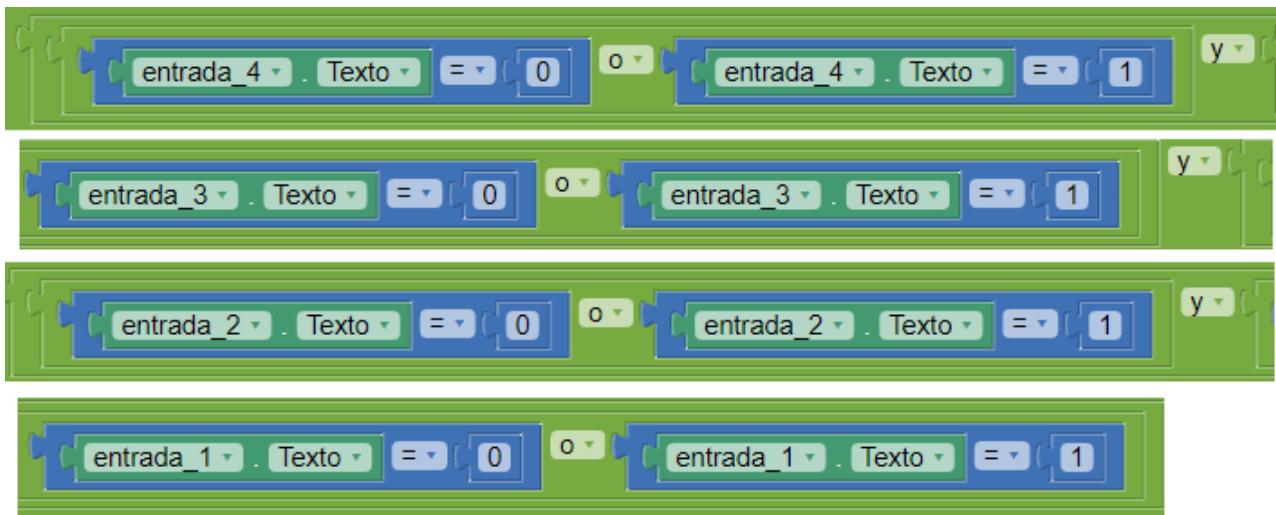
sino

Este ciclo realiza la validación de las condiciones, mientras se cumpla la condición Inicial se ejecuta el proceso asignación de turno caso contrario arroja el mensaje ***“El valor ingresado no es válido o se encuentra fuera de rango.”***



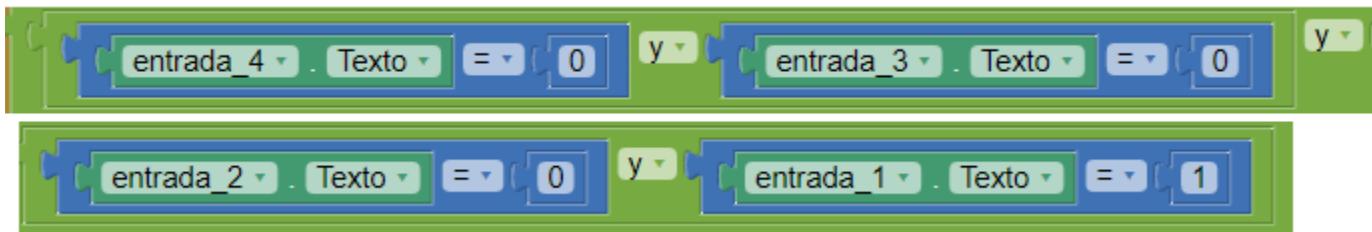
entrada_4 Texto (= 0) o entrada_4 Texto (= 1) y entrada_3 Texto (= 0) o entrada_3 Texto (= 1) y entrada_2 Texto (= 0) o entrada_2 Texto (= 1) y entrada_1 Texto (= 0) o entrada_1 Texto (= 1)

Es la condición inicial que rige el ciclo, nos indica que solo si las entradas ingresadas son “0” o “1”, el programa ingresa a un nuevo ciclo “si entonces sino” para hacer una validación final y definir las variables y configurar los mensajes que se arrojarán como resultados, caso contrario termina el ciclo e imprime el mensaje de “ERROR”



entrada_4 Texto (= 0) y entrada_3 Texto (= 0) y entrada_2 Texto (= 0) y entrada_1 Texto (= 1)

Esta es la última validación que se realiza, aquí se controla si el número ingresado es igual a “0 0 0 1” entonces se definen las variables y se imprimen los resultados correspondientes, caso contrario pasamos al siguiente número y realizamos el mismo proceso de validación siempre partiendo de un ciclo “si entonces sino”



poner TURNOS. Foto como “ seg_1.png ”

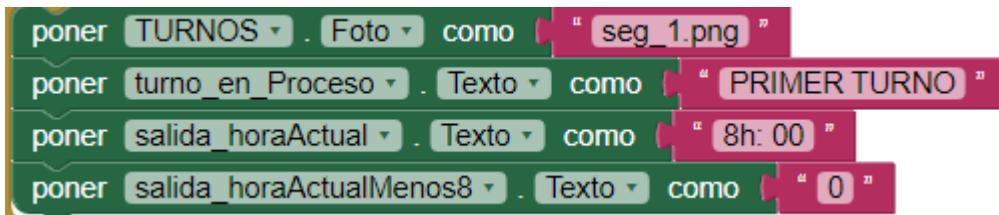
Esta variable permite imprimir la imagen que se va a proyectar según el código ingresado del turno correspondiente.

poner salida_horaActual. Texto como “8h: 00”

Esta línea de código imprime la supuesta hora actual en la que se desarrolla el turno.

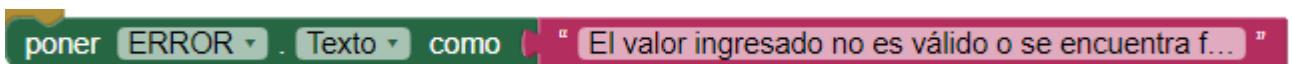
poner salida_horaActualMenos8. Texto como “0”

Esta línea de código imprime el resultado de restar la supuesta hora actual menos 8.



poner ERROR Texto como “El valor ingresado no es válido o se encuentra fuera de rango”

Con esta variable se manda a imprimir un mensaje de advertencia en el caso que los valores ingresados por el usuario no sean los correctos.



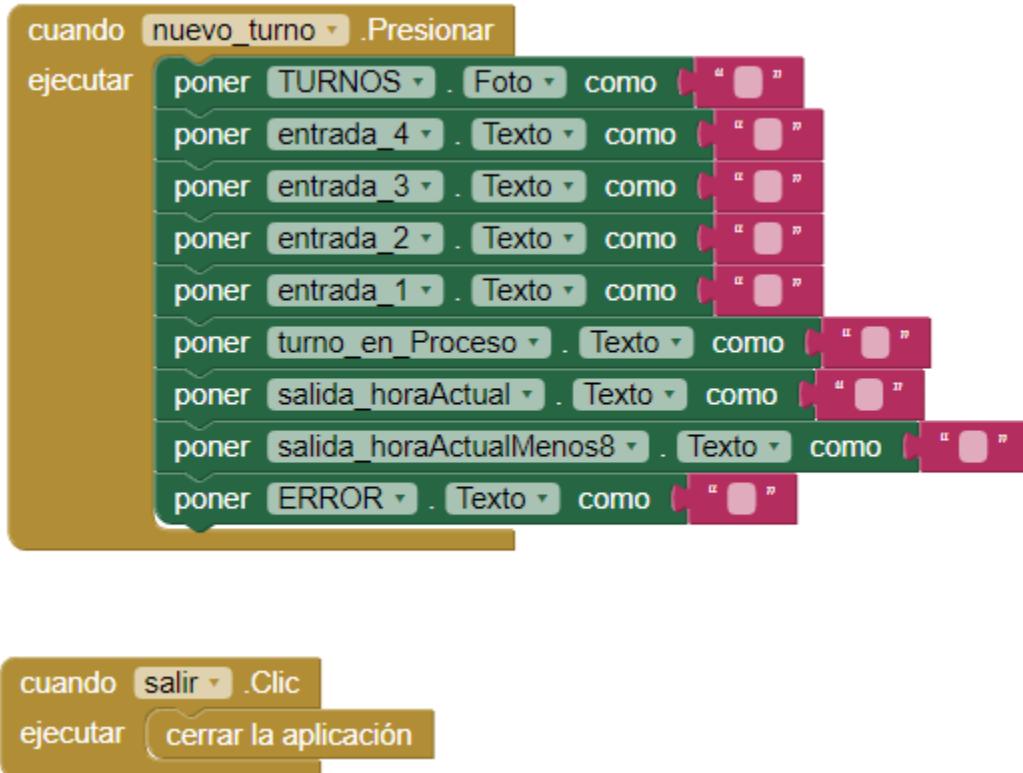
2. cuando nuevo_turno. Presionar

ejecutar

Este proceso reinicia todas las variables, limpiando la pantalla para un nuevo cálculo.

3. cuando salir. Clic ejecutar cerrar la aplicación

Este proceso termina el programa y cierra la aplicación.



9.DESCRIPCIÓN DE PRERREQUISITOS Y CONFIGURACIÓN

Tener una cuenta de Google activa es esencial, debido a que la verificación y creación de la cuenta en APP INVENTOR y TINKERCAD será a partir de esta.

Es esencial tener una conexión estable a internet, debido que será necesario para la creación del programa en app inventor.

Los sistemas operativos de los computadores en los cuales funciona app inventor son:

- Macintosh (con procesador Intel): Mac OS X 10.5, 10.6
- Windows: Windows XP, Windows Vista, Windows 7
- GNU / Linux: Ubuntu 8 +, 5 + Debian

Tener un computador con todas las actualizaciones necesarias, como también nuestro navegador deberá tener las siguientes especificaciones:

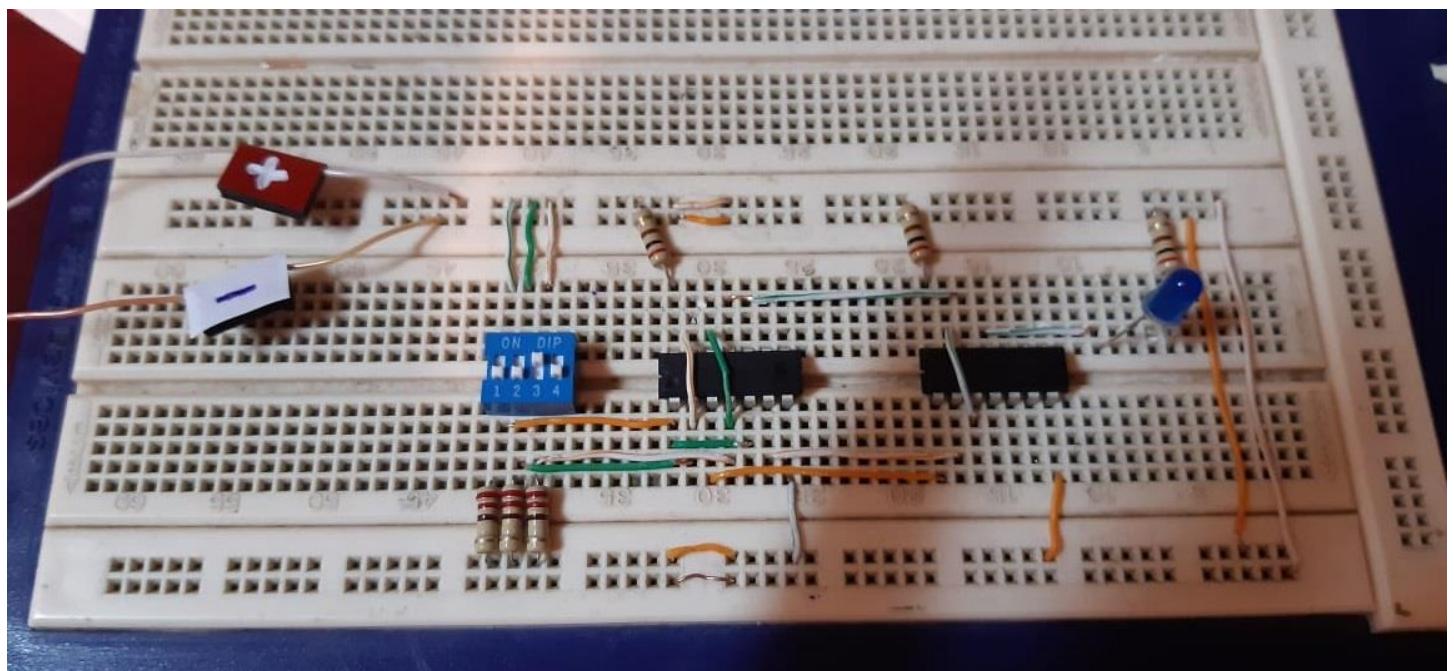
- Mozilla Firefox 3.6 o superior
- Apple Safari 5.0 o superior
- Google Chrome 4.0 o superior

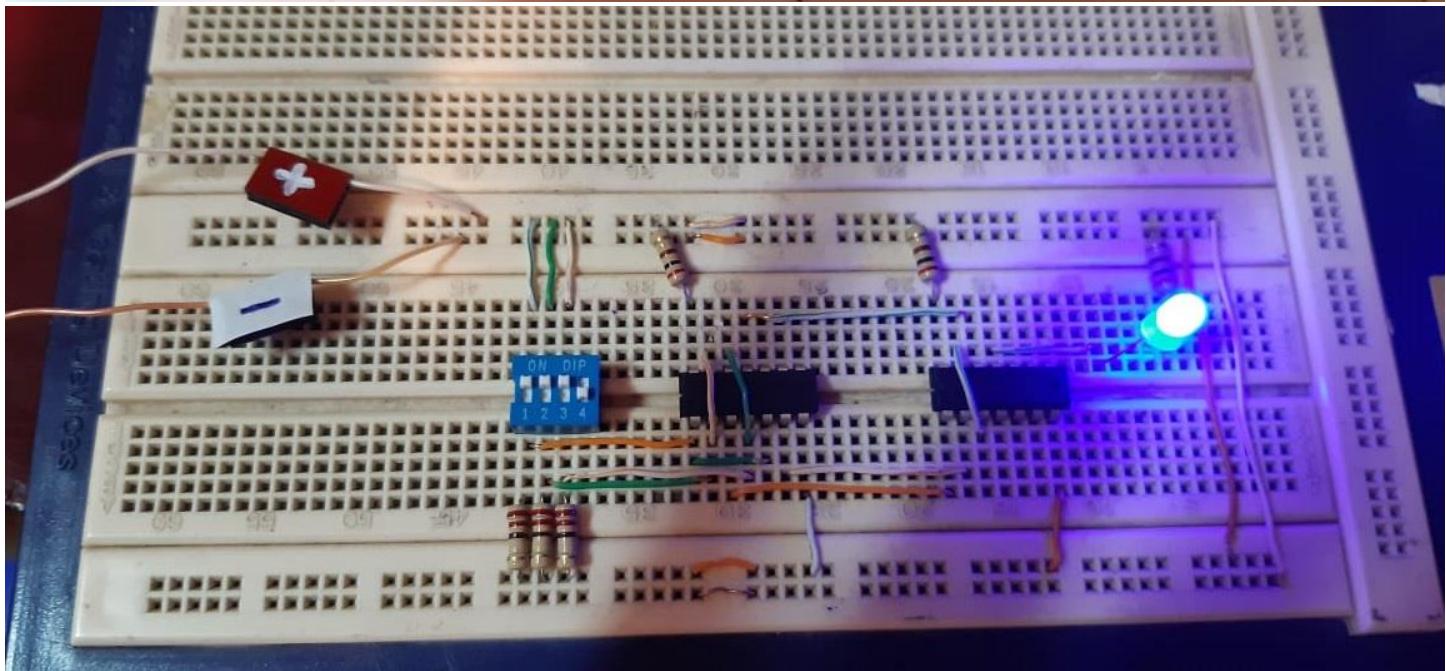
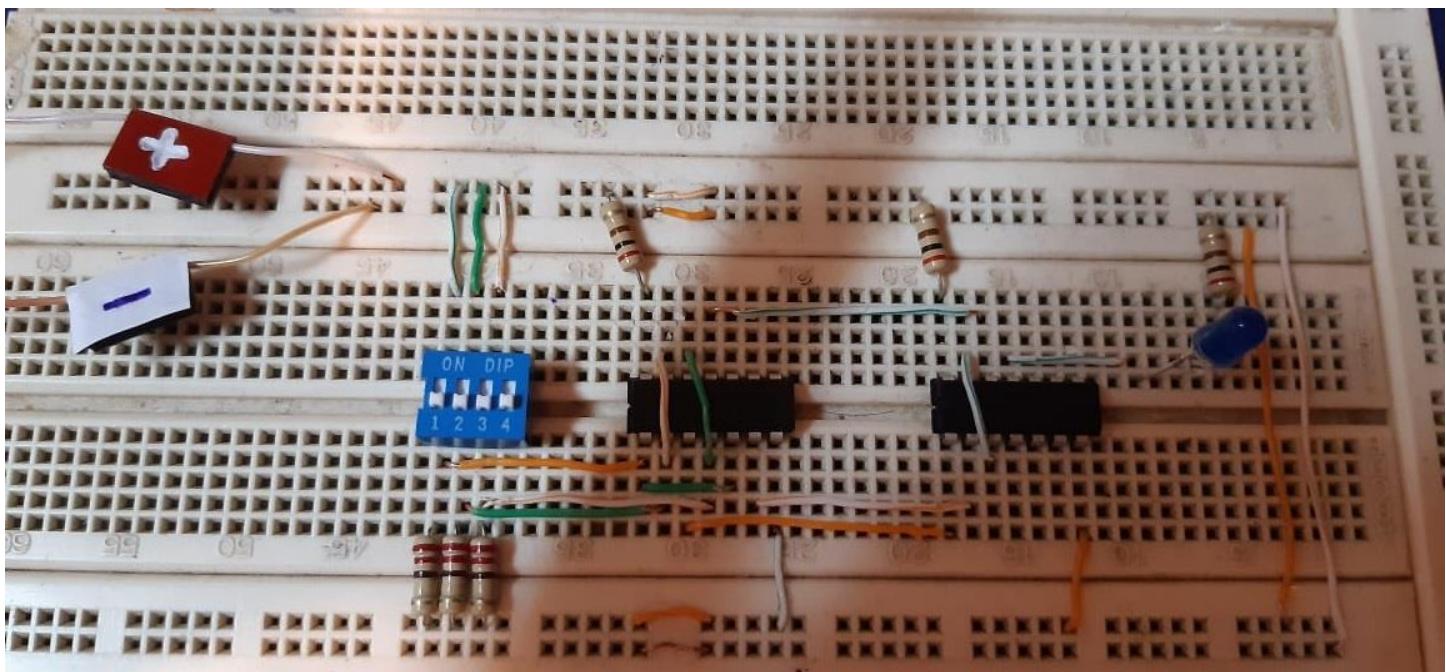
La aplicación creada con App Inventor puede funcionar en cualquier teléfono Android. El entorno de desarrollo y software de instalación se apoya directamente a los teléfonos siguientes: Los dispositivos adicionales requieren los controladores de Windows proporcionados por el fabricante del hardware.

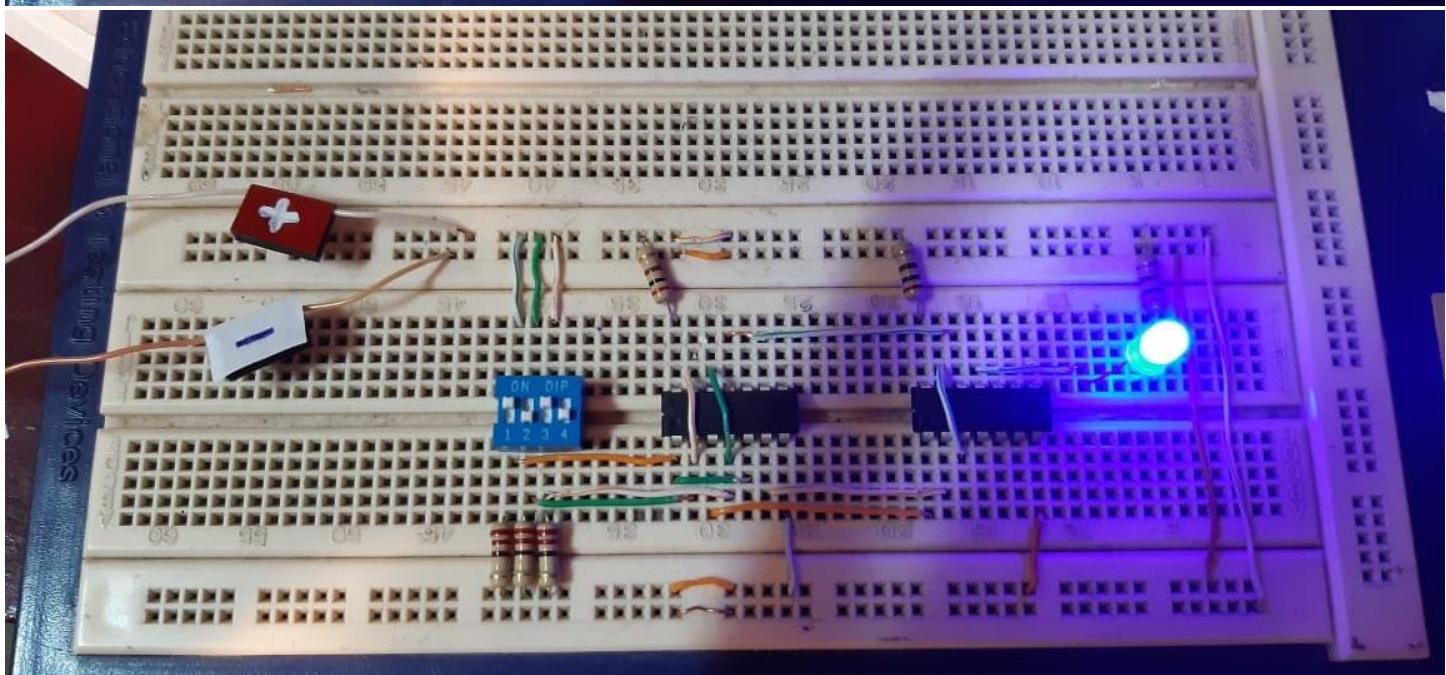
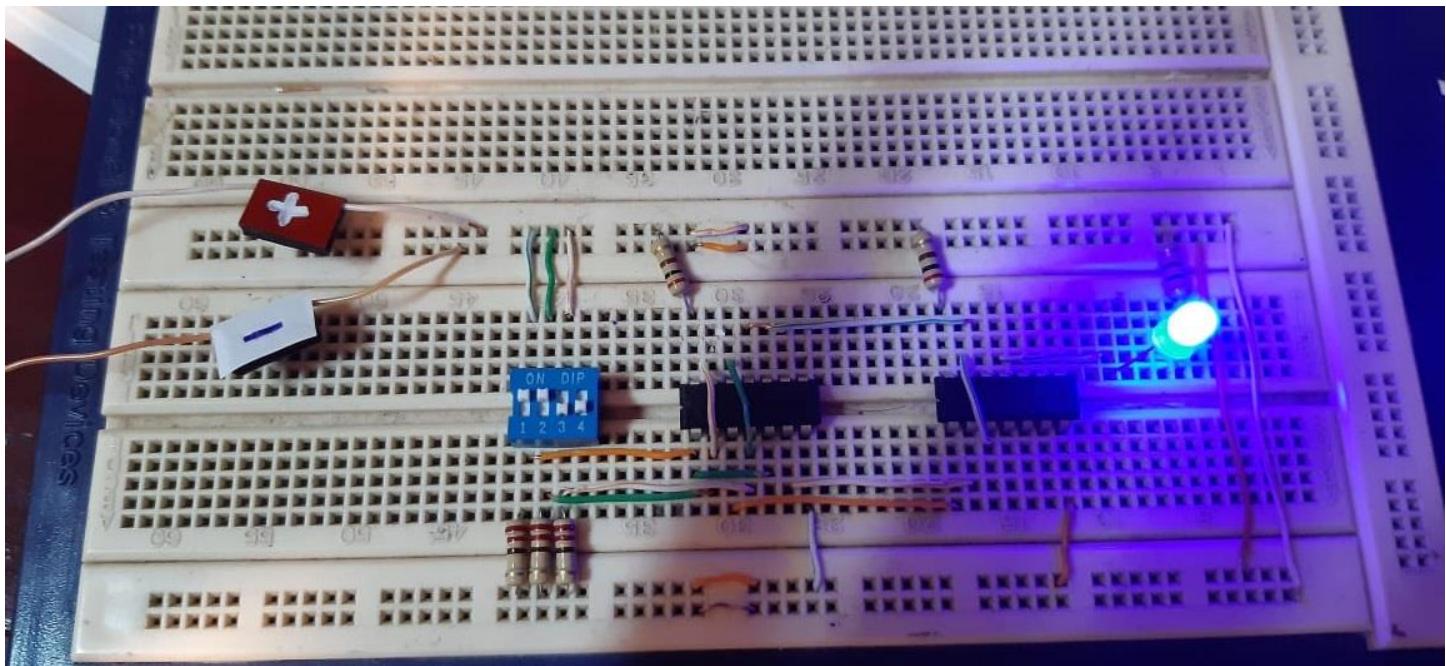
Para realizar los test de nuestro programa en el móvil, será necesario descargarnos una app desde play store para Android. El nombre de la app es “MIT AI2 Companion”

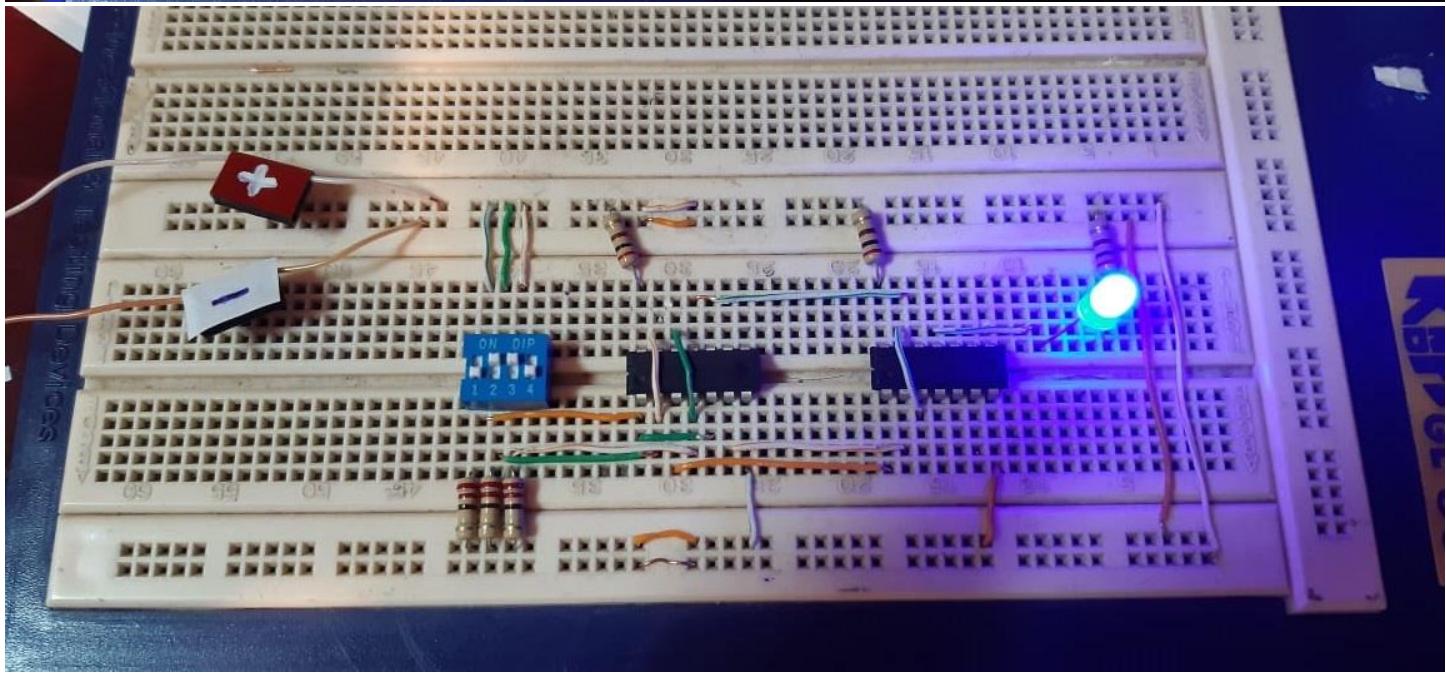
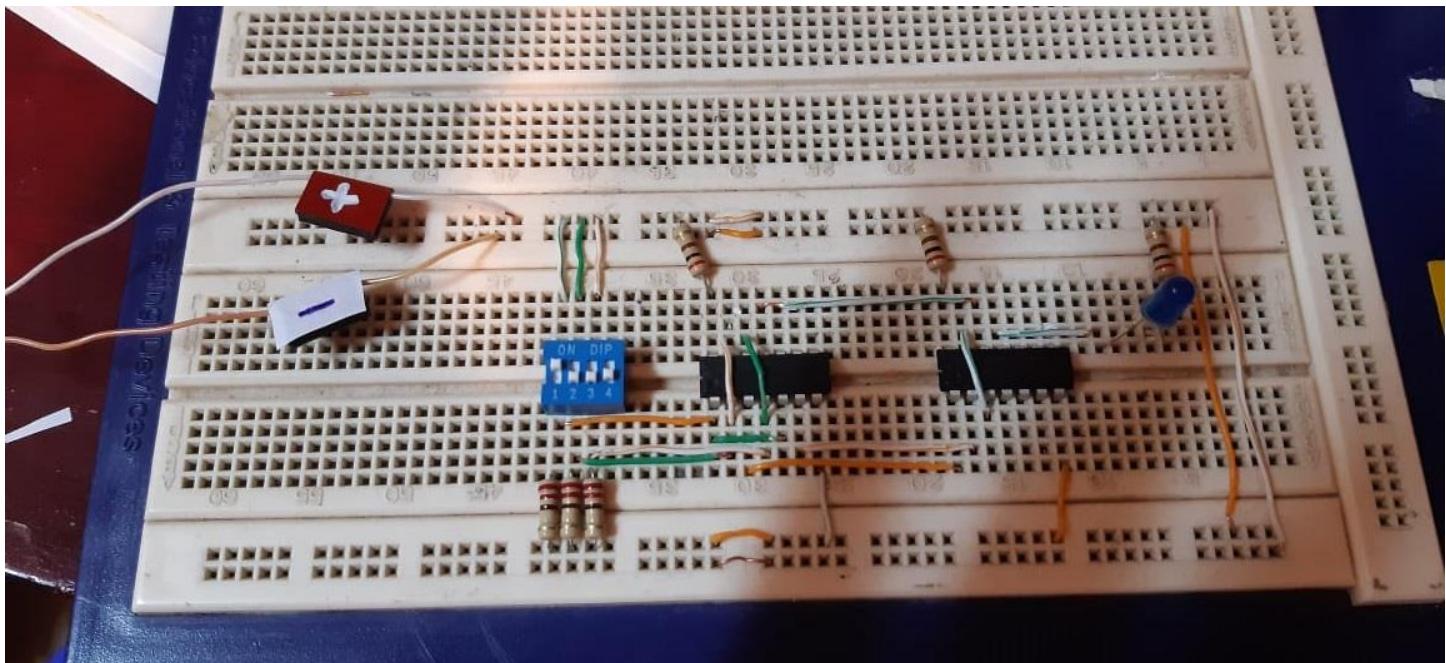
10. APORTACIONES

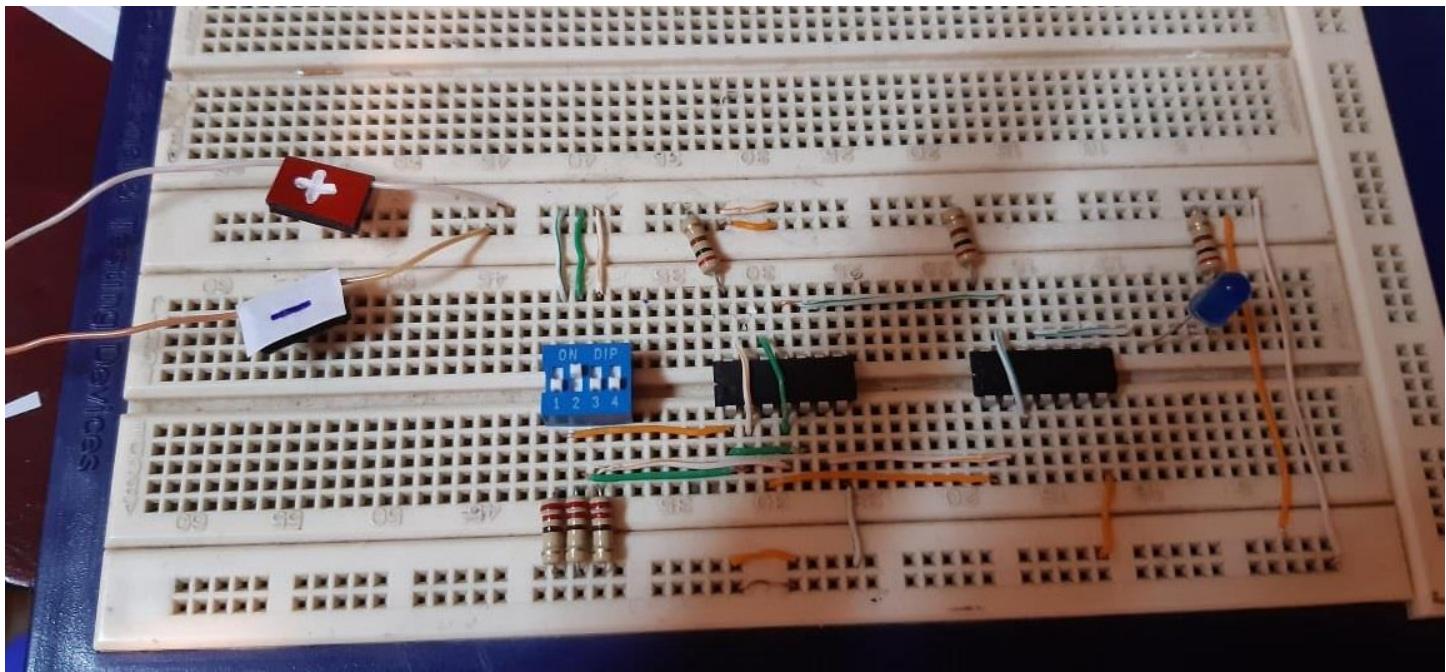
Implementación en físico del circuito votador











11.CONCLUSIONES

En conclusión:

- El circuito diseñado utilizado como norma de seguridad de aviones modernos tiene tres señales de entrada que son las salidas triplicadas así si un circuito llegara a fallar quedarían dos en buen estado, evitándose una catástrofe.
- Dado un problema que conlleva realizar una o varias acciones se lo puede resolver fragmentándolo en una o varias funciones lógicas implementadas en un circuito combinacional, estas funciones por lo general son expresiones extensas según el problema o el número de entradas o salidas que se necesite, a estas expresiones se las pueden simplificar en una expresión más trabajada por medio de aplicar los teoremas del Álgebra Booleana o los Mapas de Karnaugh , siendo este último el más óptimo para simplificar expresiones extensas, basado en la resolución del ejercicio número 2 se puede llegar a una función simplificada solo aplicando los teoremas del Álgebra Booleana .
- Se comprendió con claridad el funcionamiento de las compuertas lógicas y su implementación en los diferentes integrados. También se analizó las diferencias al simplificar los términos de forma manual y mediante los mapas de Karnaugh, observando que al usar los mapas de Karnaugh, es mucho más efectivo y rápido de implementar. En general sirvió para comprender el funcionamiento básicos de los aparatos electrónicos, como también para manejar de mejor manera el Álgebra de Boole.

12.RECOMENDACIONES

- Es de vital importancia conocer el funcionamiento de un circuito votador y las puertas lógicas básicas que se podrían utilizar.
- Al momento de plantear las variables del problema no generar variables innecesarias ya que lo único que se va a conseguir con ello es generar una función más extensa de lo realmente necesario, una vez establecida la tabla el método más recomendable para hallar la función es fijarse en la cantidad de “0” y “1” que tiene la o las salidas de esta forma se puede implementar min-términos si la cantidad de “1” es menor y máx-términos si la cantidad de “0” es menor, esto facilitará en gran medida los cálculos posteriores.
- Es importante tener claro el diseño eléctrico antes de iniciar su implementación, recordando que es necesario comprobar que los dispositivos electrónicos,en este caso los integrados, se encuentren en buen estado. También es importante tener claro los lugares donde serán necesarias las resistencias de protección.

13.CRONOGRAMA

Producto de unidad	Persona	Estado	Fecha	Cronograma
Análisis del producto unidad		Listo	jun-8	jun. 8 - 9
Resolución de los ejercicios propuestos.		Listo	jun-10	jun. 10 - 14
Desarrollo del informe		Listo	jun-10	jun. 10 - 20
Implementación de los módulos adicionales		Listo	jun-12	jun. 12 - 14
Implementación en simuladores.		Listo	jun-14	jun. 14 - 17
Implementación en el laboratorio virtual		Listo	jun-14	jun. 14 - 17
Implementación en app inventor		Listo	jun-17	jun. 17 - 20
Desarrollo de los circuitos de aportación		Listo	jun-20	jun. 18 - 20
Creación y edición del video multimedia		Listo	jun-19	jun. 19 - 20

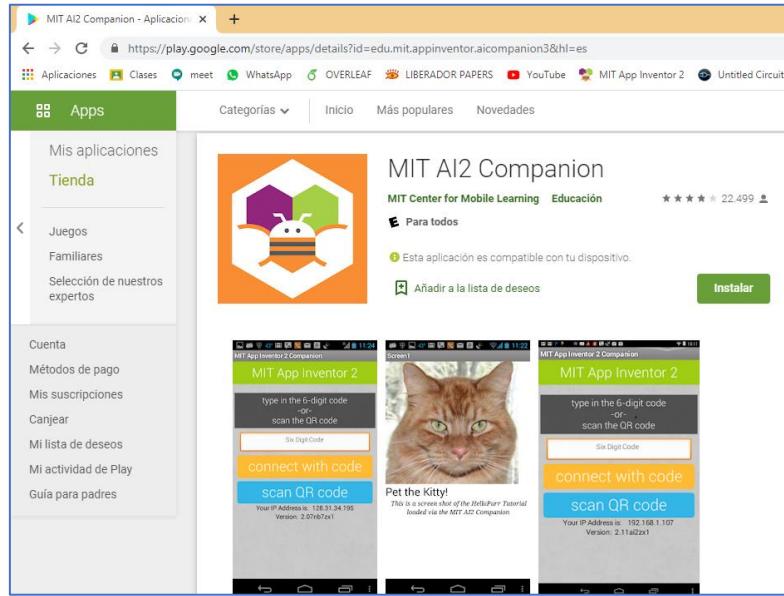
14.BIBLIOGRAFÍA

- [1] Gan, Z., Shi, G. y Shang, T. (2009). *La síntesis automática de circuitos lógicos combinacionales con algoritmo de selección clonal basado en gráficos*. 2009 Conferencia Internacional sobre Inteligencia Artificial e Inteligencia Computacional. doi: 10.1109 / aici.2009.267
- [2] Alidousti, Z. y Basiri, ME (2017). *CGACLC: Mejora del algoritmo genético a través de la agrupación para el diseño de circuitos lógicos combinacionales*. 2017 Tercera Conferencia Internacional sobre Reconocimiento de Patrones y Análisis de Imágenes (IPRIA). doi: 10.1109 / pria.2017.7983062
- [3] Mukherjee, B. y Dandapat, AK (2010). *Diseño de circuitos combinacionales mediante el método combinacional cíclico para VLSI de baja potencia*. Simposio internacional 2010 sobre diseño de sistemas electrónicos. doi: 10.1109 / ised.2010.29
- [4] Xu, Y., Ouyang, D., Ye, Y. y He, J. (2010). *Solución de problemas SAT con álgebra booleana*. Quinta Conferencia Internacional 2010 sobre la frontera de la informática y la tecnología. doi: 10.1109 / fcst.2010.13
- [5] Huang, R. y Turbak, F. (2019). Un diseño para la conversión bidireccional entre bloques y texto para App Inventor. 2019 IEEE Blocks and Beyond Workshop (B&B). doi: 10.1109 / bb48857.2019.8941197
- [6] Gonzales,Erick,(2018). Display 7 Segmentos ánodo y cátodo común.Recuperado de:
<https://hetpro-store.com/TUTORIALES/display-7-segmentos-anodo-catodo-comun/>
- [7] Posada.Fernando,(2019). Creando aplicaciones para móviles Android con MIT App Inventor 2. DOI (web) 104438/2695-4176_OTE_2019_847-19-121-5. <https://intef.es/wp-content/uploads/2019/03/MIT-App-Inventor-2.pdf>

15.ANEXOS

15.1. MANUAL DE USUARIO ENFOCADO EN EL ENUNCIADO DOS

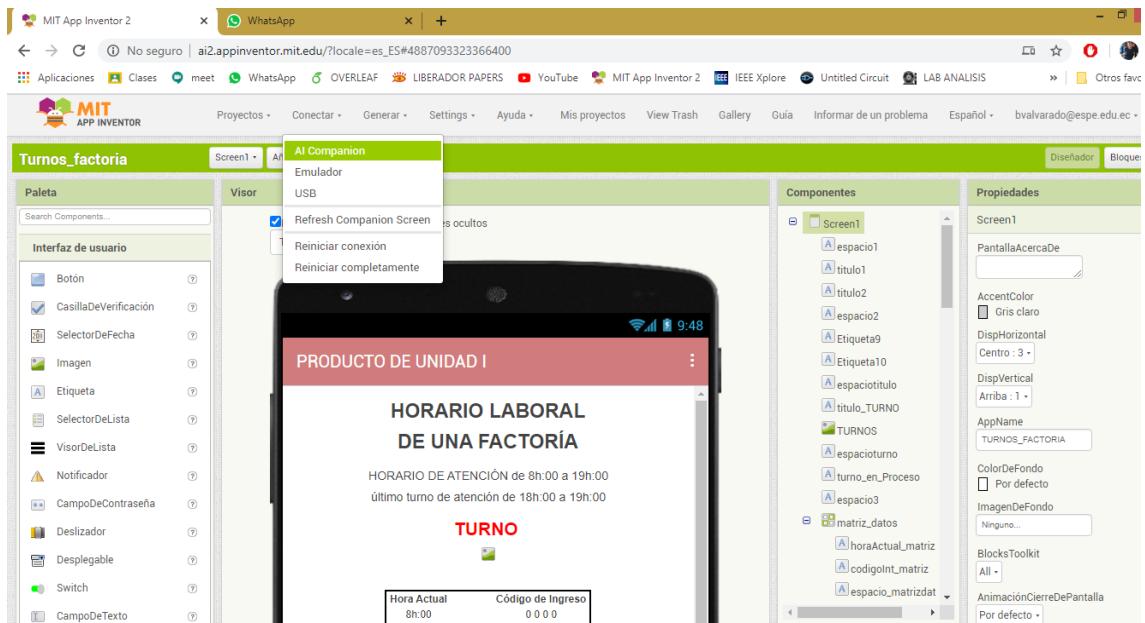
1. Para ejecutar la aplicación Turnos_Factoria, primero se debe instalar el apk MIT AI2 Companion en nuestro Smartphone con sistema ANDROID 6.0 en adelante, la descarga del apk se la realiza directamente del Play Store o en el enlace: <https://play.google.com/store/apps/details?id=edu.mit.appinventor.aicompanion3&hl=es>



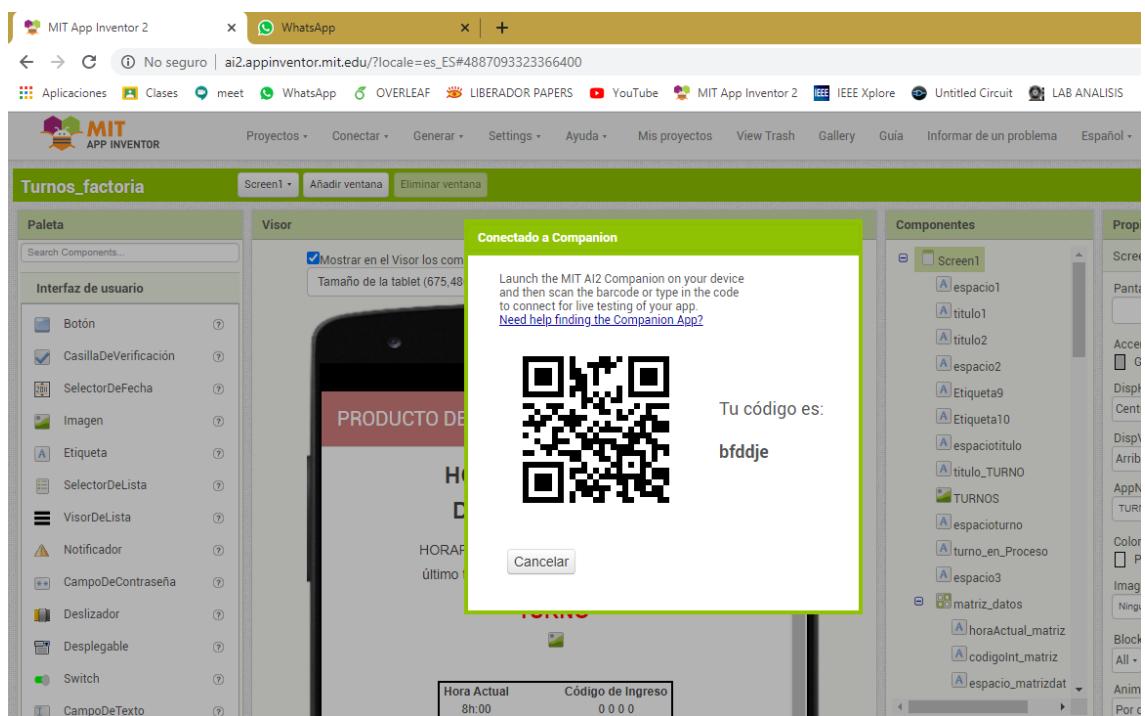
- Desde un ordenador dirigirse al sitio web de App Inventor seleccionamos el proyecto Turnos_factoria en el enlace: http://ai2.appinventor.mit.edu/?locale=es_ES#4887093323366400

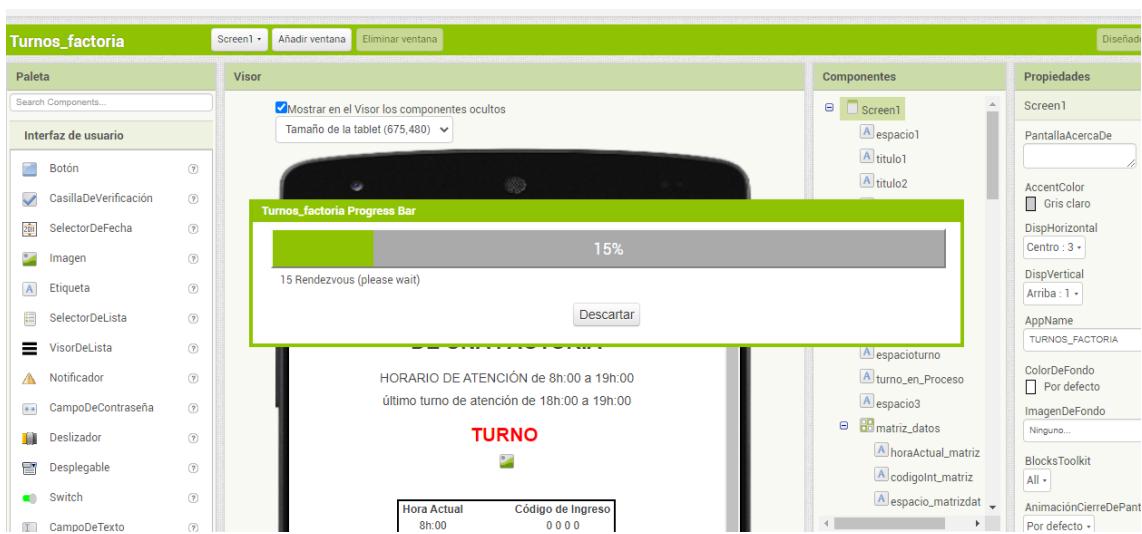
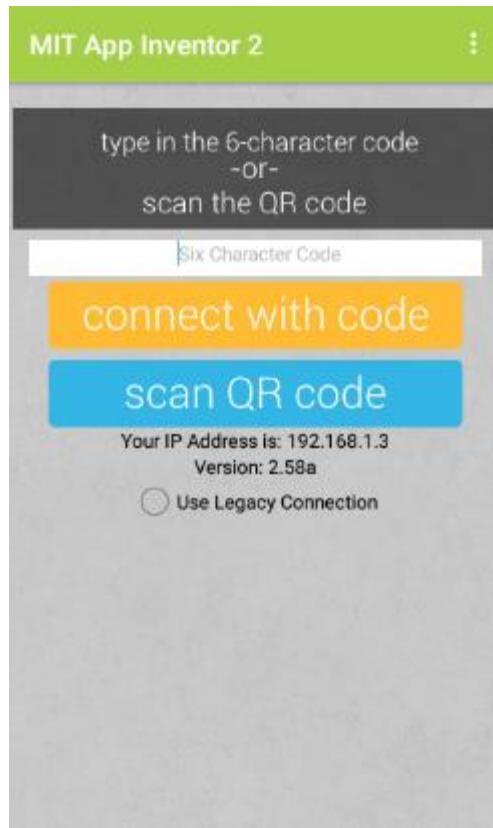
Nombre	Fecha de creación	Fecha modificación	Published
<input type="checkbox"/> Turnos_factoria	21/6/2020 6:09:18	22/6/2020 0:30:44	No
<input type="checkbox"/> Trab_Investigacion1_ALVARADO_CRUZ_HIDALGO	14/6/2020 20:00:10	15/6/2020 1:58:12	No

- Se puede visualizar la interfaz gráfica del proyecto, dar clic sobre la pestaña Conectar / AI Companion



4. Aparece un código QR como paso siguiente se debe abrir la aplicación MIT App Inventor 2 en el celular y escanear dicho código dando clic en la opción “scan QR code” y esperamos que se vincule nuestro dispositivo con el sitio web.





5. Automáticamente la aplicación se abre en el celular, en el campo “ingrese el código respectivo para realizar una consulta” se puede ingresar “0” o “1”

PRODUCTO DE UNIDAD I

:

PARA REALIZAR SU CONSULTA

entrada 4 entrada 3 entrada 2 entrada 1

Nota: el formato de la "HORA ACTUAL" está dado por numeración decimal sin minutos. Ejm: 8h:00 = "8"

CONSULTAR TURNO

NUEVO TURNO

HORA ACTUAL

(HORA ACTUAL) - 8

- Una vez ingresado el valor deseado le damos clic en CONSULTAR TURNO, se puede observar que la aplicación muestra tanto la HORA ACTUAL como la (HORA ACTUAL – 8)

PRODUCTO DE UNIDAD I

:

INGRESE EL CÓDIGO RESPECTIVO PARA REALIZAR SU CONSULTA

0 1 0 0

Nota: el formato de la "HORA ACTUAL" está dado por numeración decimal sin minutos. Ejm: 8h:00 = "8"

CONSULTAR TURNO

NUEVO TURNO

HORA ACTUAL

(HORA ACTUAL) - 8

12h: 00

4

- Finalmente, para ingresar un nuevo número, se debe dar clic en el botón "NUEVO TURNO" caso contrario dar clic en la opción "SALIR" para cerrar la aplicación.

PRODUCTO DE UNIDAD I

:

PARA REALIZAR SU CONSULTA

entrada 4 entrada 3 entrada 2 entrada 1

Nota: el formato de la "HORA ACTUAL" está dado por
numeración decimal sin minutos. Ejm: 8h:00 = "8"

CONSULTAR TURNO

NUEVO TURNO

HORA ACTUAL

(HORA ACTUAL) - 8

SALIR

15.2. HOJAS TÉCNICAS

BEAM PENTODE
FOR AF POWER-AMPLIFIER APPLICATIONS

DESCRIPTION AND RATING

The 7408 is a beam-power pentode primarily designed for use in audio-frequency power-amplifier applications. The tube is a direct replacement for the 6V6-GT, but features additional controlled zero-bias characteristics.

GENERAL
ELECTRICAL

Cathode—Coated Unipotential

Heater Voltage, AC or DC*	6.3 ± 0.6	Volts
Heater Current†	0.45	Amperes
Direct Interelectrode Capacitances‡		
Grid-Number 1 to Plate: (g1 to p)	0.7	pF
Input: g1 to (h+k+g2+b.p.)	9.0	pF
Output: p to (h+k+g2+b.p.)	7.5	pF

MECHANICAL

Mounting Position—Any

Envelope—T-9, Glass

Base—B7-7, Intermediate-Shell Octal 7-Pin or

B7-59, Short Intermediate-Shell Octal 7-Pin with carriers

MAXIMUM RATINGS
DESIGN-MAXIMUM VALUES

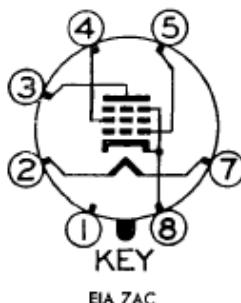
Plate Voltage.....	.350	Volts
Screen Voltage.....	.315	Volts
Plate Dissipation.....	14	Watts
Screen Dissipation.....	2.2	Watts
Heater-Cathode Voltage		
Heater Positive with Respect to Cathode		
DC Component.....	100	Volts
Total DC and Peak.....	200	Volts
Heater Negative with Respect to Cathode		
Total DC and Peak.....	200	Volts
Grid-Number 1 Circuit Resistance		
With Fixed Bias.....	0.1	Megohms
With Cathode Bias.....	0.5	Megohms

Design-Maximum ratings are limiting values of operating and environmental conditions applicable to a bogey electron tube of a specified type as defined by its published data and should not be exceeded under the worst probable conditions.

The tube manufacturer chooses these values to provide acceptable serviceability of the tube, making allowance for the effects of changes in operating conditions due to variations in the characteristics of the tube under consideration.

The equipment manufacturer should design so that initially and throughout life no design-maximum value for the intended service is exceeded with a bogey tube under the worst probable operating conditions with respect to supply-voltage variation, equipment component variation, equipment control adjustment, load variation, signal variation, environmental conditions, and variations in the characteristics of all other electron devices in the equipment.

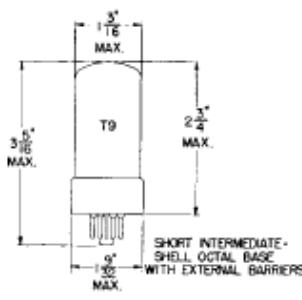
The tubes and arrangements disclosed herein may be covered by patents of General Electric Company or others. Neither the disclosure of any information herein nor the sale of tubes by General Electric Company conveys any license under patent claims covering combinations of tubes with other devices or elements. In the absence of an express written agreement to the contrary, General Electric Company assumes no liability for patent infringement arising out of any use of the tubes with other devices or elements by any purchaser of tubes or others.

BASING DIAGRAM


EIA 7AC

TERMINAL CONNECTIONS

- Pin 1—No Connection
- Pin 2—Heater
- Pin 3—Plate
- Pin 4—Grid Number 2 (Screen)
- Pin 5—Grid Number 1
- Pin 7—Heater
- Pin 8—Cathode and Beam Plates

PHYSICAL DIMENSIONS


EIA 9-41

GENERAL ELECTRIC

DM74LS32

Quad 2-Input OR Gate

General Description

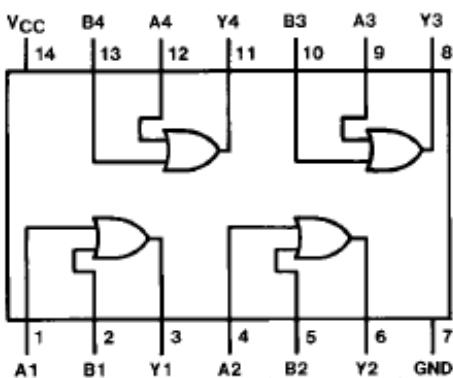
This device contains four independent gates each of which performs the logic OR function.

Ordering Code:

Order Number	Package Number	Package Description
DM74LS32M	M14A	14-Lead Small Outline Integrated Circuit (SOIC), JEDEC MS-120, 0.150 Narrow
DM74LS32SJ	M14D	14-Lead Small Outline Package (SOP), EIAJ TYPE II, 5.3mm Wide
DM74LS32N	N14A	14-Lead Plastic Dual-In-Line Package (PDIP), JEDEC MS-001, 0.300 Wide

Devices also available in Tape and Reel. Specify by appending the suffix letter "X" to the ordering code.

Connection Diagram



Function Table

$$Y = A + B$$

Inputs		Output
A	B	Y
L	L	L
L	H	H
H	L	H
H	H	H

H = HIGH Logic Level
L = LOW Logic Level

PIN DESCRIPCIÓN

PIN NO.	SÍMBOLO	NOMBRE Y FUNCION
1, 3, 9	1A a 3A	entradas de datos
2, 4, 10	1B a 3B	entradas de datos
7	GND	tierra (0 V)
12, 6, 8	1Y a 3Y	salidas de datos
13, 5, 11	1C a 3C	entradas de datos
14	V _{cc}	tensión de alimentación positiva

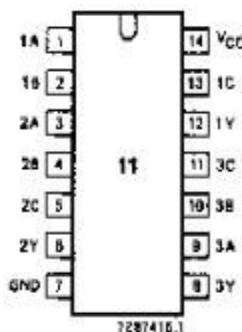


Fig.1 Configuración del pin.



Fig.2 Simbolo lógico.

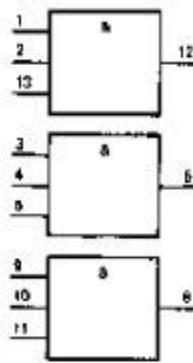


Fig.3 Simbolo lógico IEC.

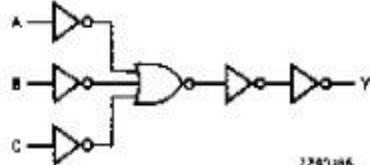
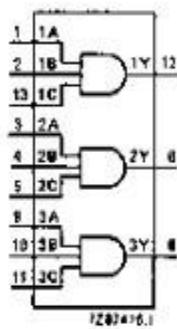


TABLA DE FUNCIONES

ENTRADAS		SALIDA	
n/A	nótese bien Carolina dell'Isola York	nótese bien Carolina dell'Isola York	nótese bien Carolina dell'Isola York
LLL			L
LL		H	L
LHL			L
LH		H	L
HL		L	L
HLH			L
S.S		L	L

ORDERING INFORMATION

TYPE NUMBER	PACKAGE				
	TEMPERATURE RANGE	PINS	PACKAGE	MATERIAL	CODE
74HC04N	-40 to +125 °C	14	DIP14	plastic	SOT27-1
74HCT04N	-40 to +125 °C	14	DIP14	plastic	SOT27-1
74HC04D	-40 to +125 °C	14	SO14	plastic	SOT108-1
74HCT04D	-40 to +125 °C	14	SO14	plastic	SOT108-1
74HC04DB	-40 to +125 °C	14	SSOP14	plastic	SOT337-1
74HCT04DB	-40 to +125 °C	14	SSOP14	plastic	SOT337-1
74HC04PW	-40 to +125 °C	14	TSSOP14	plastic	SOT402-1
74HCT04PW	-40 to +125 °C	14	TSSOP14	plastic	SOT402-1
74HC04BQ	-40 to +125 °C	14	DHVQFN14	plastic	SOT762-1
74HCT04BQ	-40 to +125 °C	14	DHVQFN14	plastic	SOT762-1

PINNING

PIN	SYMBOL	DESCRIPTION
1	1A	data input
2	1Y	data output
3	2A	data input
4	2Y	data output
5	3A	data input
6	3Y	data output
7	GND	ground (0 V)
8	4Y	data output
9	4A	data input
10	5Y	data output
11	5A	data input
12	6Y	data output
13	6A	data input
14	V _{CC}	supply voltage

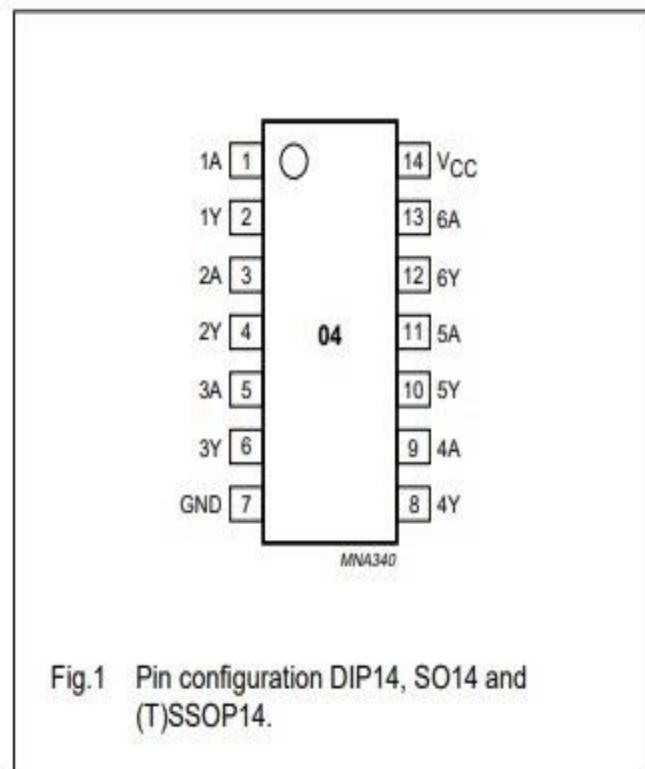


Fig.1 Pin configuration DIP14, SO14 and (T)SSOP14.

DM74LS02

Quad 2-Input NOR Gate

General Description

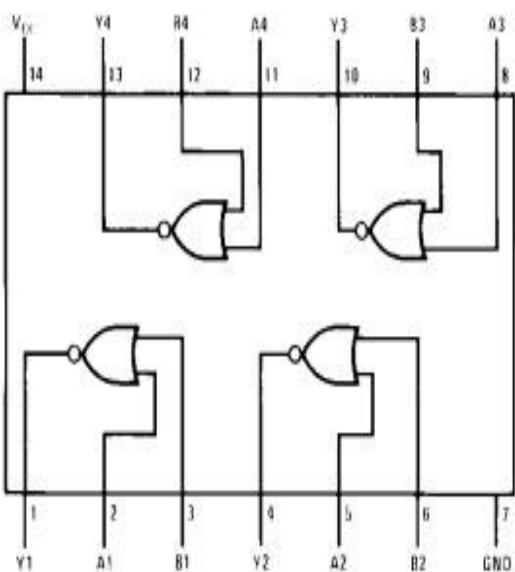
This device contains four independent gates each of which performs the logic NOR function.

Ordering Code:

Order Number	Package Number	Package Description
DM74LS02M	M14A	14-Lead Small Outline Integrated Circuit (SOIC), JEDEC MS-120, 0.150 Narrow
DM74LS02SJ	M14D	14-Lead Small Outline Package (SOP), EIAJ TYPE II, 5.3mm Wide
DM74LS02N	N14A	14-Lead Plastic Dual-In-Line Package (PDIP), JEDEC MS-001, 0.300 Wide

Devices also available in Tape and Reel. Specify by appending the suffix letter "X" to the ordering code.

Connection Diagram



Function Table

$$Y = \overline{A + B}$$

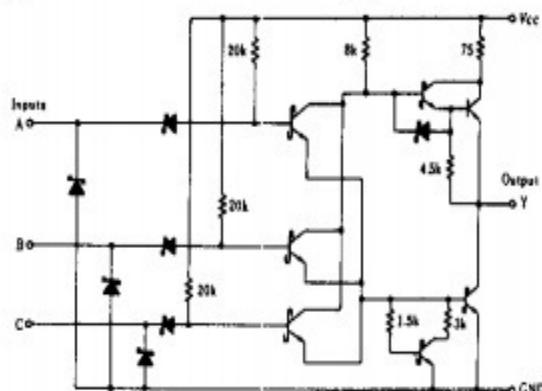
Inputs		Output
A	B	Y
L	L	H
L	H	L
H	L	L
H	H	L

H = HIGH Logic Level

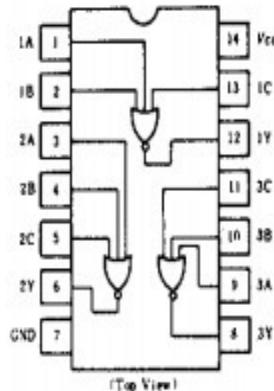
L = LOW Logic Level

HD74LS27 • Triple 3-input Positive NOR Gates

CIRCUIT SCHEMATIC(1/3)



PIN ARRANGEMENT



ELECTRICAL CHARACTERISTICS ($T_a = -20 \sim +75^\circ\text{C}$)

Item	Symbol	Test Conditions	min	typ*	max	Unit
Input voltage	V_{IH}		2.0	—	—	V
	V_{IL}		—	—	0.8	
Output voltage	V_{OH}	$V_{CC} = 4.75\text{V}, V_{IL} = 0.8\text{V}, I_{OH} = -400\mu\text{A}$	2.7	—	—	V
	V_{OL}	$V_{CC} = 4.75\text{V}, V_{IH} = 2\text{V}$	—	—	0.4	
Input current	I_{IH}	$V_{CC} = 5.25\text{V}, V_I = 2.7\text{V}$	—	—	20	μA
	I_{IL}	$V_{CC} = 5.25\text{V}, V_I = 0.4\text{V}$	—	—	-0.4	mA
	I_I	$V_{CC} = 5.25\text{V}, V_I = 7\text{V}$	—	—	0.1	mA
Short-circuit output current	I_{OS}	$V_{CC} = 5.25\text{V}$	-20	—	-100	mA
Supply current	I_{CCH}	$V_{CC} = 5.25\text{V}$	—	2.0	4.0	mA
	I_{CCL}	$V_{CC} = 5.25\text{V}$	—	3.4	6.8	mA
Input clamp voltage	V_{IK}	$V_{CC} = 4.75\text{V}, I_{IN} = -18\text{mA}$	—	—	-1.5	V

* $V_{CC} = 5\text{V}, T_a = 25^\circ\text{C}$

SWITCHING CHARACTERISTICS ($V_{CC} = 5\text{V}, T_a = 25^\circ\text{C}$)

Item	Symbol	Test Conditions	min	typ	max	Unit
Propagation delay time	t_{PLH}	$C_L = 15\text{pF}, R_L = 2\text{k}\Omega$	—	10	15	ns
	t_{PHL}		—	10	15	