**INFORME**

1. **PLANTEAMIENTO DEL PROBLEMA**

Resolver 3 problemas puntuales, ayudar a la seguridad de un avión diseñando un circuito votador que escoja la salida de 3 circuitos si es que uno de esta falla, poder saber que turno tiene asignado un trabajador ingresando la hora en código binario y por último poder comparar dos números de 2 bits mediante el diseño de un circuito comparador. Todos estos diseños serán implementados en un simulador y laboratorio virtual, adicionalmente para el caso del circuito votador se lo implementará también en una aplicación desarrollada en la plataforma de app inventor.

1. **OBJETIVOS**
   1. **Objetivo general**

* Diseñar un circuito votador, un circuito que en su salida nos indique el numero de turno ingresando la hora en código binario y un circuito digital comparador de 2 números de 2 bits.

**2.2 Objetivo específicos**

* Implementar los 3 circuitos en un simulador y un laboratorio virtual.
* Diseñar una aplicación en app Inventor del primer problema que tenga las mismas características.
* Implementar a su salida un display de 7 segmentos a los dos últimos problemas planteados

1. **ESTADO DEL ARTE**

Según (Santos, 2018) en su publicación “Una interpretación lógico-matemática para una dualidad kaingang” estudio la dualidad de compuertas donde nos indica mediante el álgebra de Boole puede llegar hacer operaciones para llegar a una diferente función pero que cumple una misma tabla de verdad donde aplica en un paso la ley de Morgan que nos indica dándonos la posibilidad de llevar a otras expresiones que necesitemos en este presente documento se usara estas leyes y toda el álgebra de Boole.

1. **MARCO TEÓRICO**

Para comenzar tenemos que tener claro las siguientes definiciones

1. **DIAGRAMAS**

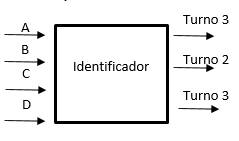


Figura 1. Diagrama de Bloques problema 1

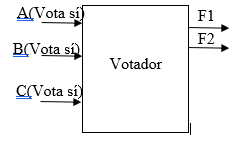


Figura 2. Diagrama de Bloques problema 2



Figura 3. Diagrama de Bloques problema 3

1. **LISTA DE COMPONENTES**

* App inventor
* Simulador Multisim
* Laboratorio virtual Tinkercad

1. **MAPA DE VARIABLES**

|  |  |  |
| --- | --- | --- |
| Circuito Votador | Circuito de turno | Comparador |
| A: Primer votador dice que sí. | A b c d | El numero A: |
| B: Segundo votador dice que sí. | Turno 1 (de 0 al 2 en decimal) | El número B: |
| C: Tercer votador dice que sí. | Turno 2 (de 3 al 4 en decimal) | M = 1 si A>B |
| F1: Los votadores resuelven que si se escoge ese circuito. | Turno 3 (de 8 al 10 en decimal) | 1= 1 si A=B |
| F2: Los votantes resuelven que no se escoge ese circuito. |  | m = 1 si A<B |

***Tabla 1*** Variables de entrada y salida

1. **DISEÑO DE LOS CIRCUITOS Y EXPLICACIÓN DEL CÓDIGO FUENTE**

**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.**

**Circuito Votador**

**Diseño**

Como tenemos 3 entradas para los 3 votos que necesitamos.

Entonces serán 2n combinaciones para nuestra tabla de verdad, donde para nuestro caso n=3.

2n=23=8

**Declaración de variables**

A: Primer votador dice que sí.

B: Segundo votador dice que sí.

C: Tercer votador dice que sí.

F1: Los votadores resuelven que si se escoge ese circuito.

F2: Los votantes resuelven que no se escoge ese circuito.

**Tabla de verdad**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| A | B | C | F1 | F2 |
| 0 | 0 | 0 | 0 | 1 |
| 0 | 0 | 1 | 0 | 1 |
| 0 | 1 | 0 | 0 | 1 |
| 0 | 1 | 1 | 1 | 0 |
| 1 | 0 | 0 | 0 | 1 |
| 1 | 0 | 1 | 1 | 0 |
| 1 | 1 | 0 | 1 | 0 |
| 1 | 1 | 1 | 1 | 0 |

***Tabla 2*** Tabla de verdad circuito de votación

**Ahora sacamos las funciones de salida utilizando los min términos**

**Para**

(Agrupamos los términos, y aplicamos )

(Aplicamos una regla de boole )

**Para**

(Ahora reducimos la función)

(Agrupamos los términos, y aplicamos )

(Aplicamos una regla de boole )

Una vez encontrado las funciones reducidas, las implementamos en el simulador multisim.

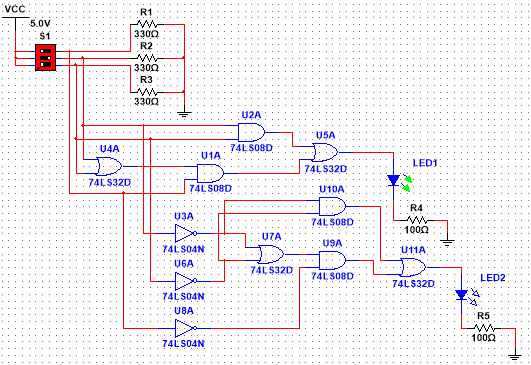
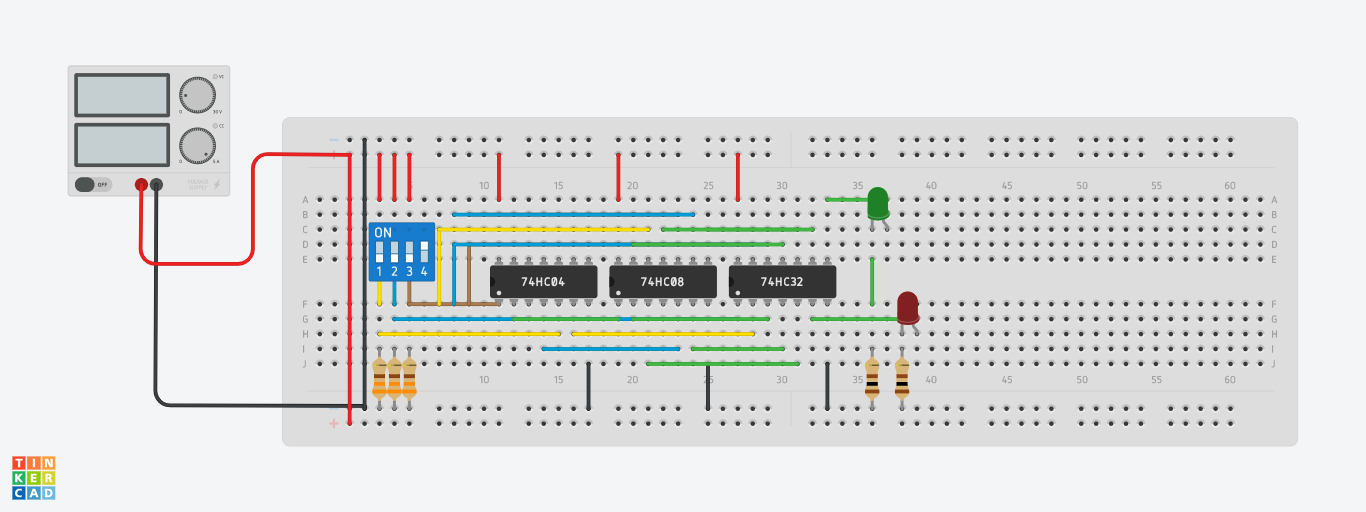


Figura 4. Circuito votador simulado en multisim

Ahora vamos a implementarlo en un laboratorio virtual en línea conocido como TinkerCad.



***Figura 5.*** Circuito votador simulado en lab virtual tinkercad

**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 las representaciones binarias de la hora actual menos ocho y que proporcione a la salida el número de turno que está trabajando si procede (si procede) o “0” si es hora de descanso**

**Variables de entrada:**

A B C D

**Variables de salida:**

Turno 1 (de 0 al 2 en decimal)

Turno 2 (de 3 al 4 en decimal)

Turno 3 (de 8 al 10 en decimal)

**Tabla de verdad**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| TURNO | HORA | HORA MENOS 8 | A | B | C | D |  | F |
| turno 1 | 8 | 0 | 0 | 0 | 0 | 0 |  | 1 |
| turno 1 | 9 | 1 | 0 | 0 | 0 | 1 |  | 1 |
| turno 1 | 10 | 2 | 0 | 0 | 1 | 0 |  | 1 |
| turno 2 | 11 | 3 | 0 | 0 | 1 | 1 |  | 1 |
| turno 2 | 12 | 4 | 0 | 1 | 0 | 0 |  | 1 |
| descanso | 13 | 5 | 0 | 1 | 0 | 1 |  | 0 |
| descanso | 14 | 6 | 0 | 1 | 1 | 0 |  | 0 |
| descanso | 15 | 7 | 0 | 1 | 1 | 1 |  | 0 |
| turno 3 | 16 | 8 | 1 | 0 | 0 | 0 |  | 1 |
| turno 3 | 17 | 9 | 1 | 0 | 0 | 1 |  | 1 |
| turno 3 | 18 | 10 | 1 | 0 | 1 | 0 |  | 1 |

***Tabla 3*** Tabla de verdad circuito de turno

Función simplificada por turnos

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| TURNO 1 |  |  |  |  |
| A'B'C'D' |  |  |  |  |
| + | A'B'C' |  |  |  |
| A'B'C'D |  |  |  |  |
| + | + | A'B'(C'+CD') | = | A'B'(C'+D') |
| A'B'CD' | A'B'CD' |  |  |  |
| TURNO 2 |  |  |  |  |
| A'B'CD |  |  |  |  |
| + | A'B'CD+A'BC'D' | A'(B'CD'+BC'D') |  |  |
| A'BC'D' |  |  |  |  |
| TURNO 3 |  |  |  |  |
| AB'C'D' |  |  |  |  |
| + | AB'C' |  |  |  |
| AB'C'D |  |  |  |  |
| + | + | AB'(C'+CD') | = | AB'(C'+D') |
| AB'CD' | AB'CD' |  |  |  |

***Tabla 4*** Simplificación de funciones

**Función implementada en proteos**

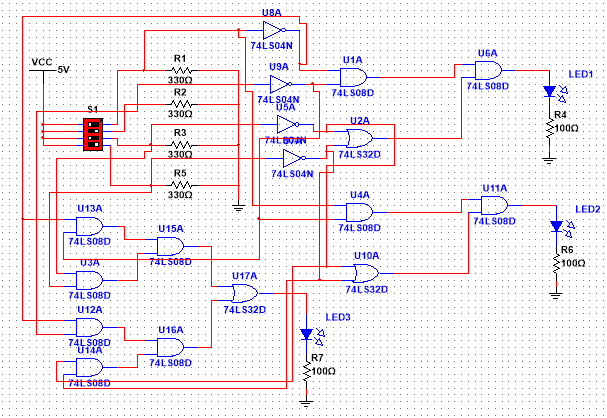


Figura 6. Circuito de turno simulado en multisim

**3. Se pretende diseñar un circuito comparador de 2 números de 2 bits, A=(a1, ao)y B=(b1,bo). Dicho circuito deberá tener tres salidas M, l, m, de tal forma que:**

**M = 1 si A>8**

**1= 1 si A=B**

**m = 1 si A<8**

**Diséñese exclusivamente con puertas NOR.**

**Variables de entrada:**

El numero A:

El número B:

**Variables de salida:**

M = 1 si A>B

1= 1 si A=B

m = 1 si A<B

**Tabla de verdad**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  | 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 |
| 0 | 1 | 1 | 1 | 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 |

**Simplificación mediante mapas K**

**A>B**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 00 | 01 | 11 | 10 |
| 00 | 0 | 1 | 1 | 1 |
| 01 | 0 | 0 | 1 | 1 |
| 11 | 0 | 0 | 0 | 0 |
| 10 | 0 | 0 | 1 | 0 |

**Función simplificada**

**Función para implementar solo con NOR**

**A=B**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 00 | 01 | 11 | 10 |
| 00 | 1 | 0 | 0 | 0 |
| 01 | 0 | 1 | 0 | 0 |
| 11 | 0 | 0 | 1 | 0 |
| 10 | 0 | 0 | 0 | 1 |

**Función simplificada**

**Función para implementar solo con NOR**

**A<B**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 00 | 01 | 11 | 10 |
| 00 | 0 | 0 | 0 | 0 |
| 01 | 1 | 0 | 0 | 0 |
| 11 | 1 | 1 | 0 | 1 |
| 10 | 1 | 1 | 0 | 0 |

**Función simplificada**

**Función para implementar solo con NOR**

**Para el tercer enunciado diseñe un módulo adicional que permita visualizar en un display de 7 segmentos conectado a cada salida el número 0 si han ocurrido uno de los 3 casos.**

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  | O |
| 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | 1 |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 0 | 1 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 0 |
| 1 | 1 | 1 | 0 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 00 | 01 | 11 | 10 |
| 0 | 0 | 1 | 0 | 1 |
| 1 | 1 | 0 | 0 | 0 |

**Función simplificada**

**Función para implementar solo con NOR**

**Simulación en multisim**

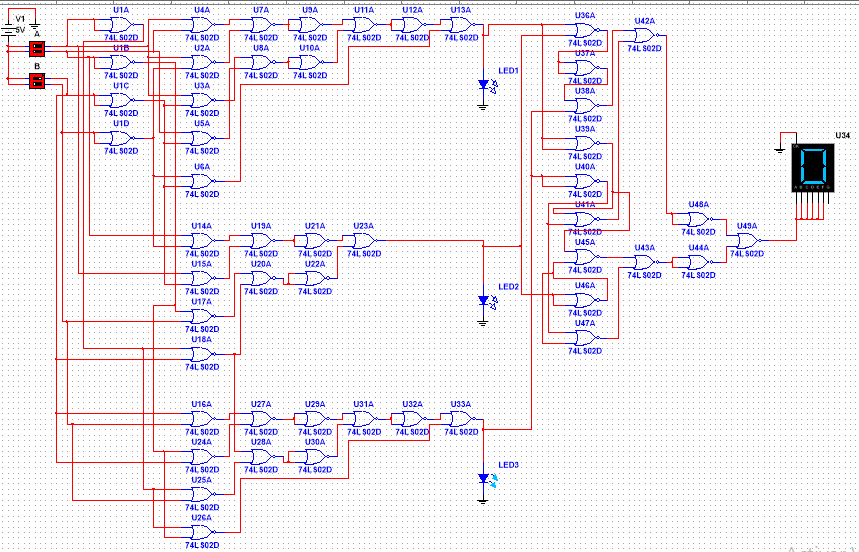


Figura 7. Circuito comparador simulado en multisim

**Simulación en laboratorio virtual**

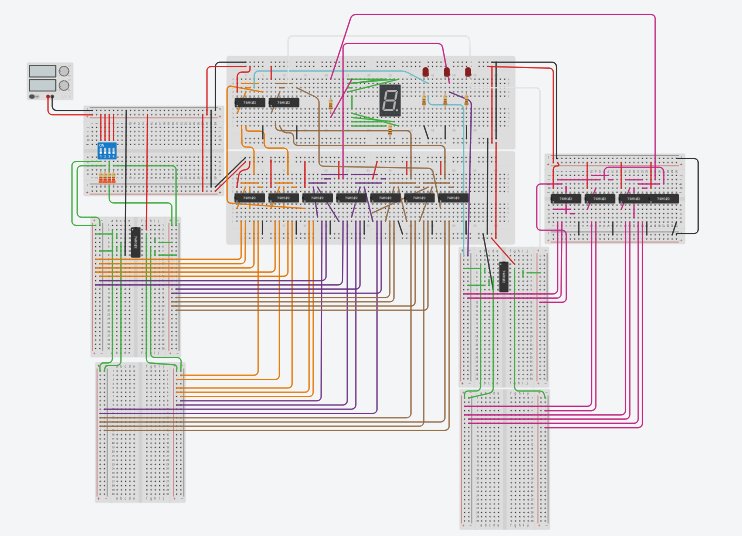
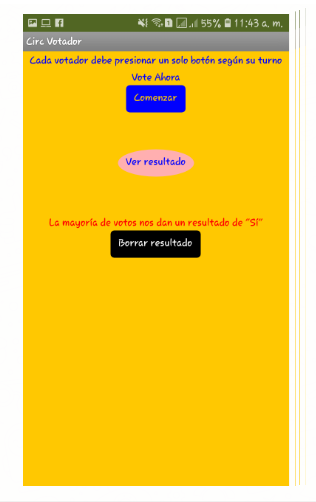


Figura 8. Circuito comparador simulado en laboratorio virtual

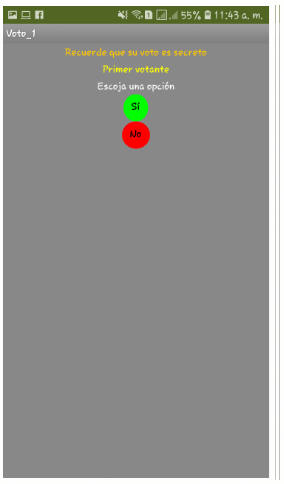
**Explicación código fuente**

**Vista Diseñador Screen1**

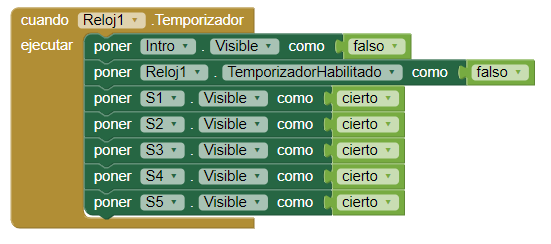


***Figura 6.*** Vista diseñador

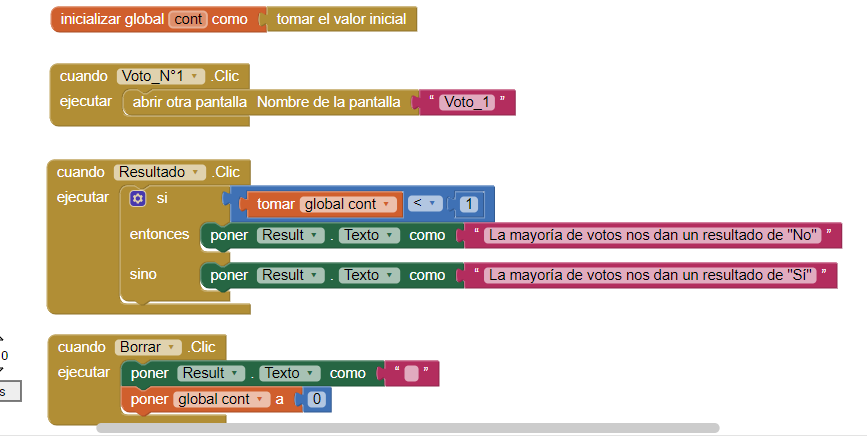
**Screen3**



***Figura 6.*** Vista de elección

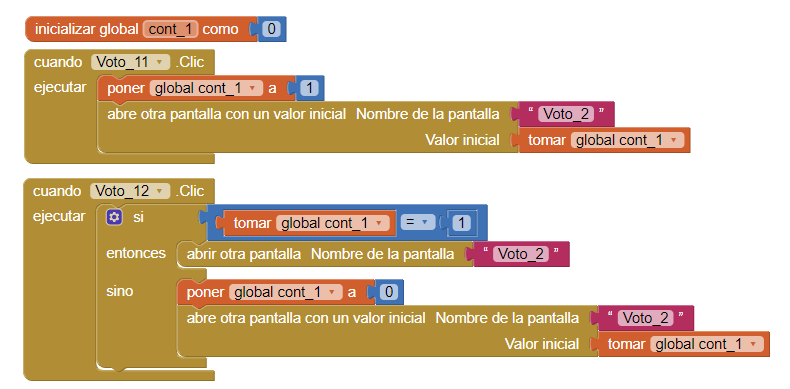


***Figura 6.*** Bloque 1

En este bloque de código armamos un splash screen, aplicando el sensor de reloj.

***Figura 6.*** Bloque 2

Ahora en este bloque se asignamos una función a cada botón de nuestro screen principal, cuando presionamos el botón comenzar nos lanza a una nueva screen en donde comenzará el primer voto, para el botón resultado verificamos el valor si es menor que 1 para contabilizar los botos y dar una sentencia de si la mayoría votó que no, caso contrario quiere decir que la mayoría de los votos fueron si, y visualizamos en un label.



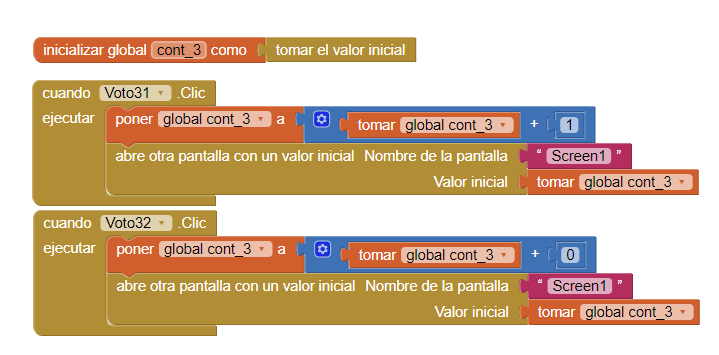
***Figura 6.*** Bloque 3

En esta parte del código nos encontramos en la screen Voto\_1 en donde inicializamos la variable cont\_1 en 0 para empezar con los votos y cuando damos clic en sí. Le asignamos un valor de 1 y si es el botón No, le asignamos un valor de cero, en cualquiera de los dos casos nos lanza a la pantalla siguiente en donde mandamos el valor de cont\_1.



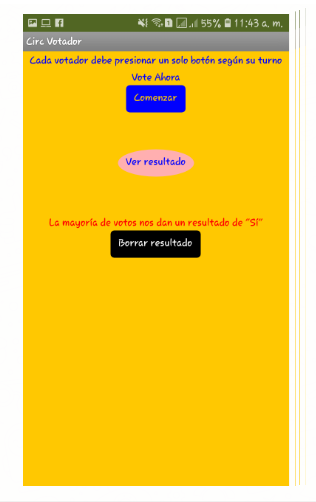
***Figura 6.*** Bloque 4

En esta parte del código nos encontramos en la screen Voto\_2 en donde inicializamos la variable cont\_2 en con el valor que mandamos desde la screen voto 1, cuando damos clic en sí, le asignamos la suma del valor que tenga nuestra varibale cont\_2 +1 y si es el botón No le sumamos 0, en cualquiera de los dos casos nos lanza a la pantalla siguiente en donde mandamos el valor de cont\_2.



***Figura 6.*** Bloque 5

En esta parte del código nos encontramos en la screen Voto\_3 en donde inicializamos la variable cont\_3 en con el valor que mandamos desde la screen voto 2, cuando damos clic en sí, le asignamos la suma del valor que tenga nuestra variable cont\_2 +1 y si es el botón No le sumamos 0, en cualquiera de los dos casos nos lanza a la pantalla principal en donde podemos ya mostrar ya los resultados con el valor que mandamos de cont\_3.



***Figura 6.*** Vista final

**9. DESCRIPCIÓN DE PRERREQUISITOS Y CONFIGURACIÓN**

Para el circuito que en su salida nos muestra el horario de turno de un trabajador, hay que tomar en cuenta que se debe ingresar la hora en código binario restándole 8 horas.

1. **APORTACIONES**

Uso de laboratorio virtual Tinkercad, este simulador es muy eficiente al momento de armar nuestros circuitos acercándose mucho a lo que es la realidad, además que posee diversos componentes para el diseño de circuitos digitales***.***

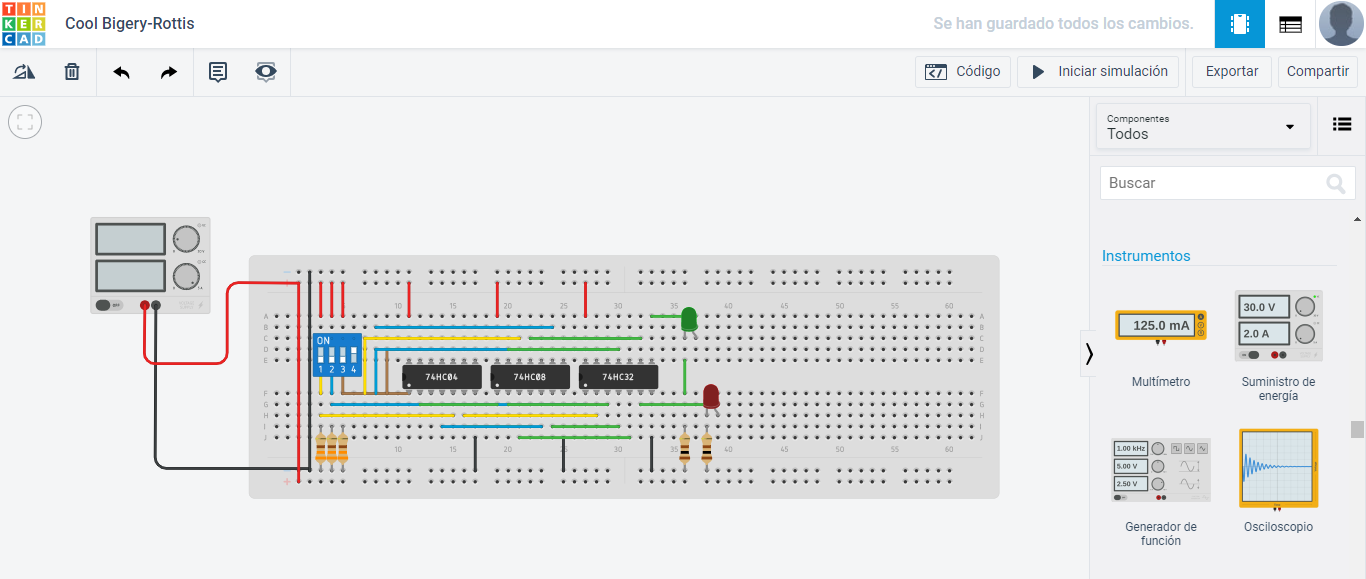


Figura 9. Circuito comparador simulado en laboratorio virtual

1. **CONCLUSIONES**

* Se diseñó el circuito votador el cual nos permite tomar una decisión para escoger una salida tomando en cuenta la mayoría de los votos, dado una falla de un sistema de seguridad de un avión
* Se implementó el ejercicio votador en app inventor
* El uso de display de 7 segmentos se puede usar con un decodificador

1. **RECOMENDACIONES**

* Se recomienda que para el uso de la App Inventor se debe tener conocimientos previos de lenguaje de programación como Java, C++, PYTHON.
* Es importante conocer el álgebra de Boole para la simplificación de funciones
* Es necesario investigar mapas K para la mejor simplificación y optimización de tiempo en simplificar funciones
* Es preciso planificar un cronograma con diagramas de Grant en las diferentes aplicaciones que existen y para el desarrollo se recomienda el software Project.

1. **CRONOGRAMA**

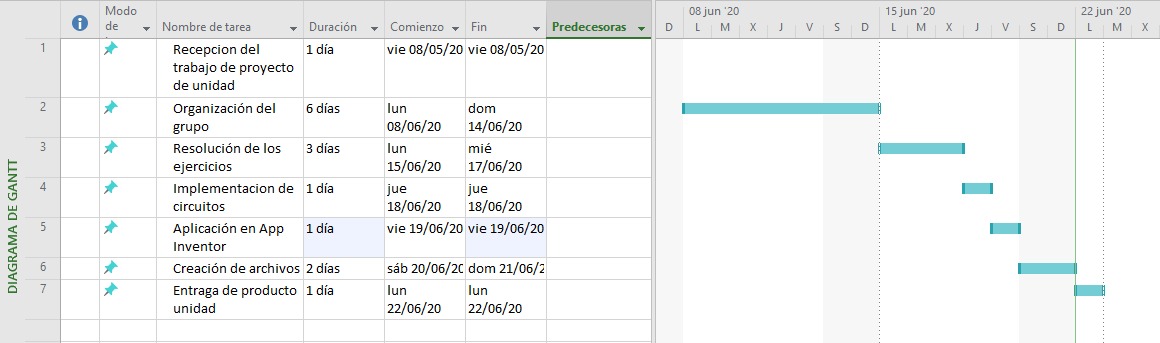
****

Figura 10. Cronograma de actividades

1. **BIBLIOGRAFÍA**

Floyd, T. (2006). *Fundamentos de sistemas digitales.* Madrid: Pearson.

Ricoy, A. (14 de Junio de 2020). *Appinventor en español*. Obtenido de https://sites.google.com/site/appinventormegusta/primeros-pasos

Santos, J. (2018). Algoritmo de las operaciones aritmeticas aplicadas a los codigos binarios, octal y hexadecimal con sus respectivas conversiones. *Revista Latinoamericana de Etnomatemática*.

1. **ANEXOS**

****

Figura 11. Interfaz de la aplicación

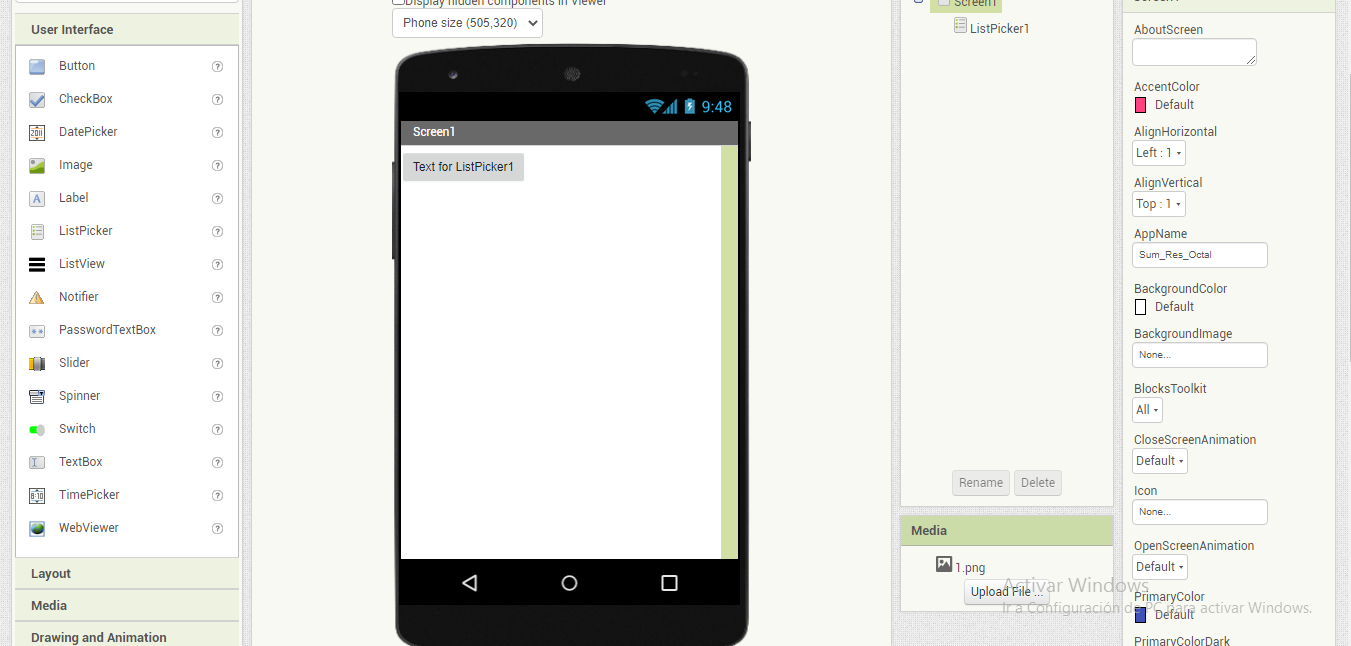


Figura 12. Entorno de trabajo parte grafica

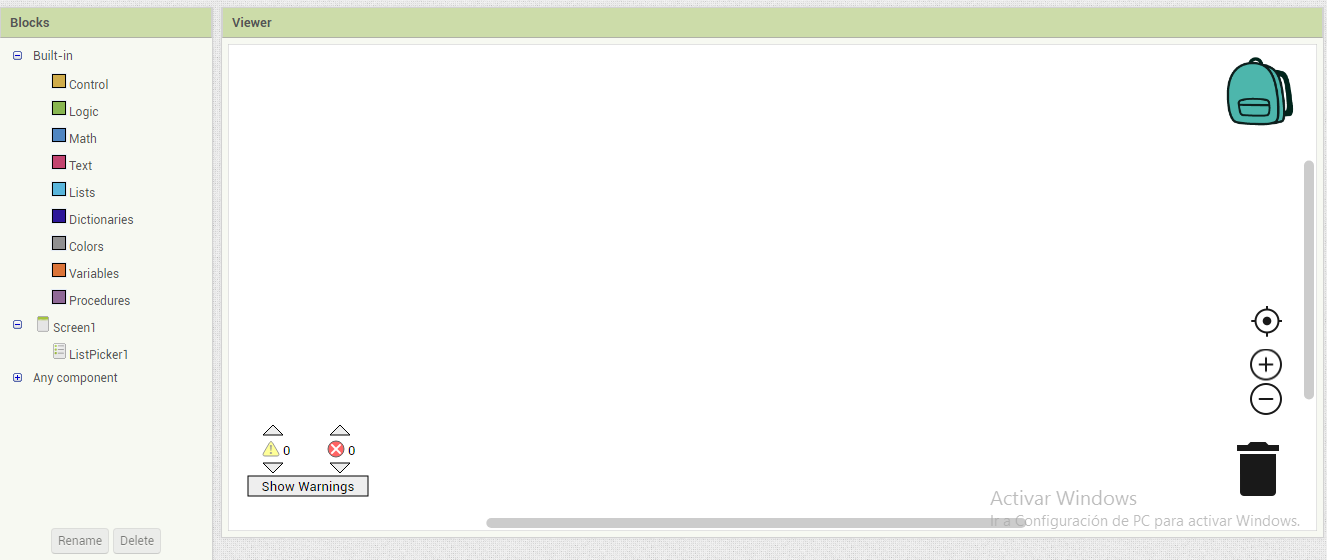


Figura 13. Entorno de trabajo parte de bloques

**15.1 MANUAL DE USUARIO**

**Circuito Votación**

La siguiente aplicación es una para contador de votos entre “sí” y “no”.

En la interfaz principal se encontrará con tres botones; uno que es “comenzar” que nos permite comenzar con el proceso de sufragar; el voto será discreto para tres votantes y se debe sufragar en orden y solo una vez por votante entre las opciones de “si” o “no”.

El botón “Ver resultado” nos permite visualizar quien obtuvo más votos entre las opciones de “si” o “no”, dándonos como mensaje “La mayoría de votos nos dan un resultado de Si” si es si el ganador y si el vencedor es no “La mayoría de votos nos dan un resultado de No”

Y para borrar los datos almacenados en ver resultados lo hacemos con el botón borrar resultado así nuestra pantalla principal quedara vacía y además de que nos permite volver a realizar el sufragio una vez más realizando el mismo proceso desde el botón comenzar.

**15.2 HOJAS TÉCNICAS**

***Ficha Técnica de Calculadora de Octal***

***Lenguaje de Programación***

|  |  |
| --- | --- |
| **Lenguaje de programación** | Programación orientada a objetos y base de datos |
| **Software de edición y programación** | App inventor II |
| **Formato de empaquetamiento** | .APK |
| **Software** | Móvil Android |

*Tabla numero 1: Especificaciones*