

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA  
FACULTAD DE INGENIERÍA  
ESCUELA DE CIENCIAS Y SISTEMAS  
ORGANIZACIÓN COMPUTACIONAL  
SEGUNDO SEMESTRE 2021  
ING. OTTO ESCOBAR  
TUTOR ACADÉMICO SECCIÓN A: EDDY SIRIN  
TUTOR ACADÉMICO SECCIÓN B: FRANCISCO LEZANA

## PROYECTO 1 BUSCAMINAS

## Objetivo General:

- Aplicar todos los conocimientos adquiridos en el curso de Organización Computacional sobre lógica combinacional y secuencial, en un proyecto que encapsule todo lo visto en el curso.

## Objetivos Específicos:

- Que el estudiante aprenda a crear circuitos secuenciales y combinacionales
- Que el estudiante aprenda la creación de la comunicación Serial.
- Que el estudiante aprenda la creación de una memoria RAM.
- Que el estudiante aprenda la aplicación de una memoria RAM.

## Descripción:

Como proyecto final del curso de Organización Computacional, se requiere que los estudiantes creen el famoso juego de Windows: Buscaminas. Este con la finalidad de aplicar todos los conocimientos adquiridos en el curso, en un mismo proyecto. Este contara de los siguientes módulos:

- Configuración del Juego
- Juego

Además de esto, se deberá realizar una parte donde se simulará un ejercicio de comunicación serial.

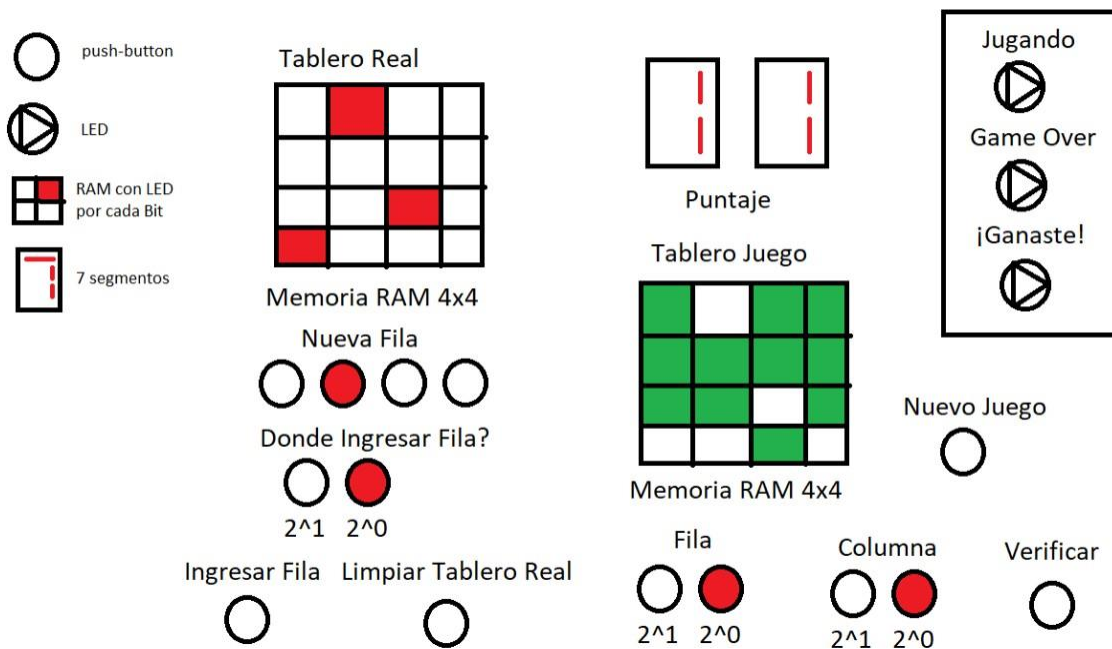


Imagen donde se explican los diferentes módulos y sus partes.

## Simulación

Se debe utilizar proteus para hacer cualquier simulación. Se recomienda el uso de puentes para tener un diseño más limpio. Se permite el uso de cualquier integrado usado en prácticas pasadas, si el circuito ya se realizó en alguna práctica, se puede usar el integrado equivalente (ejemplo, el integrado de Binario a BCD, o un controlador de stepper). Aunque si es un elemento que no se ha usado, se recomienda preguntar antes al auxiliar si está permitido su uso.

## Configuración del Juego

Para poder tener las bombas cargadas en el tablero, se necesita hacer una configuración del tablero previo a cada partida (Esto no se debe delimitar en funcionamiento, es solo en el flujo. Se configura primero y luego se juega). Para esto se tendrá una memoria RAM de 4x4 donde podremos ingresar por medio de filas un conjunto de 4 bits, donde se indicará con un "1" si hay una bomba en esa posición del tablero.

La fila y columna se debe seleccionar por medio de un selector de 2 bits, para ver en qué fila se quiere cargar el conjunto de bits. Y tendremos un botón para ingresar la fila. También se tendrá un botón para reiniciar el tablero.

Por último, se debe tener una representación con leds de la RAM, donde el led deberá encender si hay una bomba en esa posición.

**La memoria RAM debe ser creada con Flip-Flops.**

## Juego

El juego consiste básicamente en verificar todas las casillas que no tengan una bomba. A diferencia del juego original, en esta versión no hay pistas de bombas aledañas, el juego es puramente aleatorio.

El flujo de este, es ingresar la fila y columna que queremos verificar (con 2 bits de selector para cada uno), y al darle a verificar, se irá a leer esa posición en la RAM del tablero de configuración, y procede a ver si es una bomba o no, en caso sea una bomba, prende el led de gameover y se acaba la partida (no se debe hacer nada especial con esto, solo prender el led de gameover y que no estén encendidos los otros leds de "estado de juego"). En caso no sea una bomba, se ingresa un "1" en el tablero de juego, que está representado también por una RAM de 4x4. Una vez ingresado esto, aumenta el punteo del jugador en 1. Esto se repite hasta que la cantidad de 0s de la RAM del tablero de juego, sea igual a los 1s del

tablero de configuración (se prende el led de "ganaste") o se seleccione una bomba.

Los estados de juego son 3, los cuales son excluyentes. Cuando se inicia el programa, se esta jugando o empieza una nueva partida, el led se pone en estado "Jugando". Si se encuentra una bomba, se prende el led de gameover, y si se encuentran todas las posiciones sin bombas, se prende el led de ganaste. Para salir de estos 2 últimos estados hay que presionar el botón de "Nuevo Juego" el cual reseteara los LEDs a "jugando", reseteara el contador de Puntaje, y el Tablero de Juego.

## Ingreso por Comunicación Serial

MODO				
	$2^4$	$2^2$	$2^1$	$2^0$
0	0	0	0	0

Agregado al modo de juego manual, también se podrá jugar por medio de un programa (queda a su discreción el lenguaje) el cual se comunique por medio de comunicación serial al circuito implementado.

### Envío de Datos

**Posición:** Para la posición se tomará en cuenta la combinación "0000" (0) hasta "1111" (15) manejando la matriz como un arreglo lexicográfico por filas.

**Modo:** El modo servirá para definir si se posiciona una mina en el tablero cuando se envíe un "0" o si se accionará el disparador cuando se envíe un "1"

- Modo 0: Se deberá ingresar las bombas de la matriz, por medio de comunicación serial.
- Modo 1: Se podrá ingresar las posiciones al tablero de juego por mediocomunicación serial.

### Recepción de Datos

Deben guardar en la aplicación la posición actual y de esta manera solo deberán conocer si acertaron o no el tiro.

**Acierto:** Si el disparador acierta el disparo recibirán la trama "1111".

**Fallo:** Si el disparador falla el disparo recibirán la trama "0000"

## Circuito Físico

Se deberá de implementar en protoboard los siguientes circuitos:

1. **Modulo de configuración de juego**
2. Ventanas eléctricas con motor stepper, las cuales se deberá de mantenerse abiertas por 10 segundos. Estas ventanas se abrirán cada vez que se ingrese una mina en la matriz de configuración

Se deberá de implementar en placa los siguientes circuitos:

1. Matriz de led módulo de configuración
2. Contador de Punteo

## Documentación

Se deberá documentar cada parte del Proyecto. Viendo la documentación como un producto donde nosotros explicamos a alguien como funciona el dispositivo, mas enfocado a un manual técnico. Se deberán explicar los componentes utilizados en cada modulo, como funciona el modulo, tablas de verdad de alguna función realizada y screenshots del mismo circuito.

En esta se debe detallar cada modulo creado, y poner conclusiones y recomendaciones al final de la misma, para poder ver que concluyo el grupo de este proyecto.

## Observaciones y Restricciones:

- Se realizará de manera grupal.
- Copias totales o parciales tendrán una nota de 0 y serán reportadas a escuela.
- El simulador debe ser proteus.
- **NO SE PUEDE UTILIZAR LOGICSTATE PARA DAR VOLTAJE. SOLO LOGICPROBE PARA INDICAR EL RESULTADO DE LA FUNCIÓN.**
- El día de la calificación se harán preguntas sobre aspectos utilizados en la elaboración de la práctica, las cuales se considerarán en la nota final.

## Requerimientos Mínimos

- Para tener derecho a calificación:
  - o Se debe entregar la documentación
- Enviar archivo link del drive que contenga todo el código utilizado para la práctica y manual técnico antes de las 23:59 horas del viernes 5 de noviembre 2021.
  - medio: UEDi – 1 por grupo