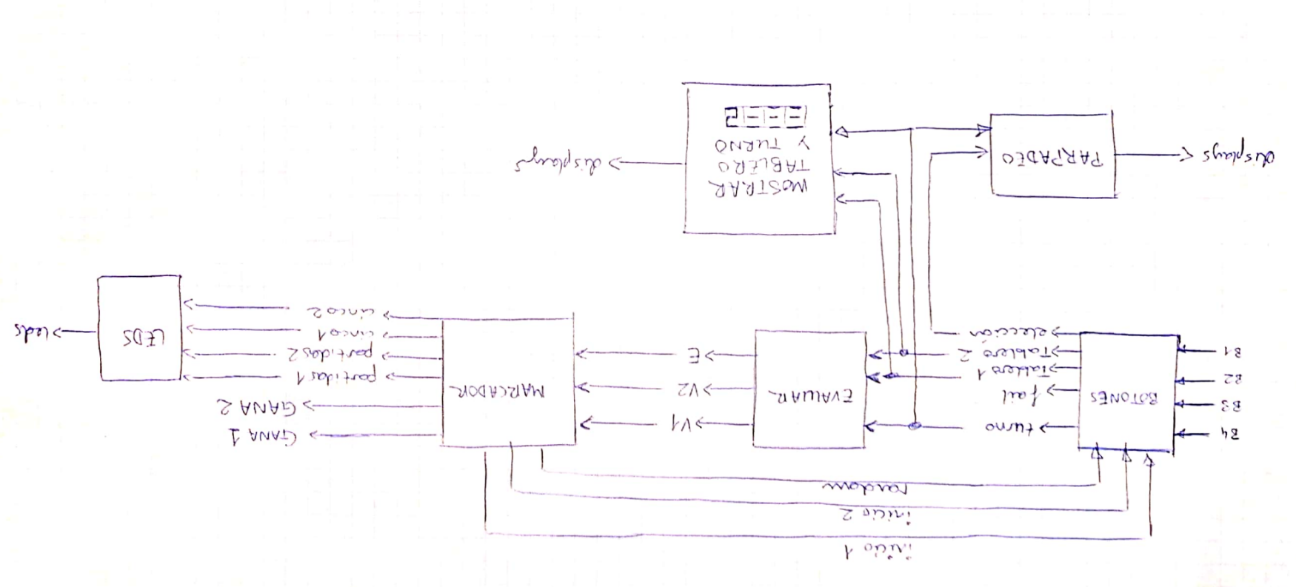
TRABAJO VHDL 3 EN RAYA

1. ESQUEMA INICIAL



1. DIAGRAMA DE BLOQUES. DESCRIPCIÓN BASICA.

**Bloque Antirebotes:** Su función es un filtrado de rebotes o estados inestables en la pulsación de los botones B1, B2, B3 y B4. Se instanciarán en el bloque botones.

*boton : in std\_logic;* : Señal de entrada externa procedente de un botón/pulsador.

*filtrado: out std\_logic;* : Señal de salida interna con la pulsación filtrada.

**Bloque Botones:** Su función es gestionar las señales procedentes de los botones para registrar la elección y confirmación de casillas para las fichas durante el juego.

Tiene como **entradas** principales, además de las señales clk y reset, las siguientes señales:

*B1: in std\_logic; B2: in std\_logic; B3: in std\_logic; B1: in std\_logic;* : Señales externas procedentes de cada uno de los pulsadores.

Tiene como **salidas** principales las siguientes señales internas:

*election: out std\_logic\_vector (8 downto 0*) : Registra la elección de la posición de la ficha en el tablero.

*B1val: out std\_logic* : Genera un pulso de reloj cuando se pulsa B1 de forma válida (después de pasar por el bloque antirrebotes), que sirve para confirmar la elección de casilla para un ficha del jugador.

El bloque contiene a su vez un bloque antirebotes para filtrar las señales de cada uno de los botones.

**Bloque Validar:** En este bloque se comprueba la validez de la casilla elegida y se actualizan los tableros, además de comprobar si se ha producido la victoria de la partida por parte de algún jugador. Como entradas fundamentales se tienen *B1val y B1prev1*, que representan la pulsación válida de B1, y *election*, que indican la selección del jugador, mientras que t*urn* indica de quién es el turno.

Como salidas principales tiene *Tablero1* y *Tablero2*, que se van actualizando, el resultado de la partida (*V1, V2, E*) y *fail,* de valor ‘1’ si la casilla elegida estaba ocupada.

**Bloque Marcador:** La funcionalidad de este bloque consiste en recoger el numero de partidas jugadas y como ha quedado la partida(victoria del jugador 1 V1, victoria del jugador 2 V2 o empate E). Como **entradas**, a parte de reset y clk, recibirá *V1,V2 y E*, indicando la situación de finalización de la partida. Dispone de **salidas** encargadas de enviar la señal de encendido de juego ganado(*gana1 y gana2*), el número de partidas jugadas(*partidas1, partidas2 y n\_partidas*) y el inicio de la siguiente partida(*inicio1, inicio2 y random en caso de empate*). Además, en caso de cumplirse un empate al final del juego, se dispondrá de un algoritmo encargado de dar ganado el juego al jugador que ganó por muerte súbita, o en caso de imposibilidad de remontar el juego, se asignará directamente al ganador sin necesidad de hacer más partidas.

**Bloque Turnos:** Actualiza el turno después de cada jugada. Las **entradas** que recibe (*fail, reset\_election, E, mode y V1*) definen las situaciones en las que hay que cambiar el turno, esto es, cada vez que hay que se haya realizado la jugada sin fallo (las dos primeras) y cada vez que sea el turno de la máquina en el modo 3. *E* de valor ‘1’ hace que el turno deba ser aleatorio. La **salida** principal es *turnk, de valor ‘0’ para el jugador 1 y ‘1’ para el jugador 2.*

**Bloque Displays:**

**Bloque Luces:**

**Bloque Temporizador:** Sirve para controlar el tiempo en el modo de funcionamiento dos. Como **entradas**, además de *clk*, *reset*, *mode* y *speed*, tiene *cambia\_turno* y *parar\_temporiador*, que reinician el temporizador y lo detienen respectivamente, y *n\_partidas*, que lleva la cuenta para controlar la variación del límite de tiempo.

La única **salida** es *fin\_tiempo*, que vale ‘1’ cuando se ha pasado el tiempo de un turno.

**Bloque Top:** Bloque superior puramente estructural en el que se estructuran el resto de bloques.