

## Pseudocódigo

- Inicio
- Verificar si el botón de inicio fue presionado
  - sí fue presionado esperamos un tiempo y volvemos a verificar esta vez si dejaron de presionar (anti rebote)
    - sí fue presionado inicia conteo de semáforo, sino no pasa nada.
      - iniciamos un ciclo que se cumplirá mientras una variable tenga un valor específico
        - revisamos si el botón de jugador 1 fue presionado con el respectivo anti rebote implementado
          - sí fue presionado verificamos si el contador del jugador 1 es =8
            - si fuese = 8 encendemos el led bandera que indica que el jugador 1 gano y cambiamos el valor de la bandera del ciclo while para que la carrera pueda iniciar de nuevo
            - si no fuese =8 cargamos un valor que predefiniremos por medio de un array en la posición que indica el contador del jugador 2 al puerto D y le sumamos 1 al contador.
        - Revisamos si el botón del jugador 2 fue presionado con el respectivo anti rebote implementado
          - Si fue presionado verificamos si el contador del jugador 1 es =8
            - Si fuese =8 encendemos el led bandera que indica que el jugador 2 gano y cambiamos el valor de la bandera del ciclo while para que la carrera pueda iniciar de nuevo
            - Si no fuese =8 cargamos un valor que predefinimos por medio de un array en la posición que indica el contador

del jugador 2 al puerto C y le sumamos 1 al contador.

- Fin.

## Código comentado

```
#include <xc.h>

//*****

// Variables

//*****

#define _XTAL_FREQ 8000000

//declaramos una variable bandera para el ciclo while del juego
unsigned char flag;

//declaramos 2 arrays que nos servirán para cargar los valores a los puertos
//de los led
int count1 [8] = {0b00000001, 0b00000010, 0b00000100, 0b00001000, 0b00010000,
    0b00100000, 0b01000000, 0b10000000};
int count2 [8] = {0b00000001, 0b00000010, 0b00000100, 0b00001000, 0b00010000,
    0b00100000, 0b01000000, 0b10000000};

//declaramos los contadores de los botonasos de los jugadores
char J1;
char J2;

//*****

// Prototipo de funciones

//*****

/*instanciamos los prototipos de las funciones a utilizar
* las cuales no devolverán nada ni pedirán parametros.
* */
```

```

void semaforo(void);

void setup(void);

/**
 *
 */

// Ciclo Principal

/**
 *
 */

void main(void) {
    //ciclo infinito
    while (1) {
        /*llamamos a la funcion que modificara los puertos para configurarlos
        como entradas o salidas y limpiarlos*/
        setup();

        //verificamos si el boton RA0 (inicio) fue presionado
        if (PORTAbits.RA0 == 0) {
            //esperamos un tiempo corto
            __delay_ms(40);

            //volvemos a verificar pero esta vez si el boton se levanto para
            //asegurarnos que efectivamente hubo un botonaso
            if (PORTAbits.RA0 == 1) {
                //si efectivamente hubo un botonaso mandamos a llamar a la
                //funcion semaforo que nos indicara cuando inicia la carrera
                semaforo();

                //le damos el valor de 1 la bandera flag
                flag = 1;

                //iniciamos el ciclo while de juego que funcionara mientras
                //flag sea =1
                while (flag == 1) {
                    //verificamos si el boton del jugador 1 fue presionado

```

```

//esto con su respectivo anti rebote como el anterior
if (PORTAbits.RA1 == 0) {
    __delay_ms(40);
    if (PORTAbits.RA1 == 1) {
        //si fue presionado verificamos si el contador del
        //jugador 1 =8
        if (J1 == 8) {
            /*si es =8 encendemos el led que indica que gana
            y salimos del ciclo de juego al ciclo principal*/
            PORTEbits.RE0 = 1;
            __delay_ms(1000);
            flag = 2;
        } else {
            //si el contador no es =8 le cargamos el valor
            //del array en de la posición J1 al puerto D
            PORTD = count1[J1];
            //y aumentamos en 1 el contador
            J1++;
        }
    }
} //verificamos si el boton del jugador 2 fue presionado
//esto con su respectivo anti rebote como el anterior
else if (PORTAbits.RA2 == 0) {
    __delay_ms(40);
    if (PORTAbits.RA2 == 1) {
        //si fue presionado verificamos si el contador del
        //jugador 2 =8

```

```

        if (J2 == 8) {
            /*si es =8 encendemos el led que indica que gana
            y salimos del ciclo de juego al ciclo principal*/
            PORTEbits.RE1 = 1;
            __delay_ms(1000);
            flag = 2;
        } else {
            //si el contador no es =8 le cargamos el valor
            //del array en de la posición J2 al puerto C
            PORTC = count2[J2];
            //y aumentamos en 1 el contador
            J2++;
        }
    }
}

}

}

}

}

}

}

}

//*****

// Configuración

//*****

void setup(void) {
    ANSEL = 0b00000000; //se deshabilitan las entradas analógicas
    ANSELH = 0b00000000;
    TRISB = 0b00000000; //configuramos el puerto B como salida

```

```

TRISA = 0b11111111; //configuramos el puerto A como entrada
TRISC = 0b00000000; //configuramos el puerto C como salida
TRISE = 0b00000000; //configuramos el puerto E como salida
TRISD = 0b00000000; //configuramos el puerto D como salida
PORTA = 0b00000000; //limpiamos los puertos para asegurarnos de empezar en 0
PORTB = 0b00000000;
PORTD = 0b00000000;
PORTC = 0b00000000;
PORTE = 0b00000000;
J1 = 0; //iniciamos los contadores en 0
J2 = 0;
}

//*****

// Funciones

//*****

void semaforo(void) {
    //encendemos el pin RB0
    PORTBbits.RB0 = 1;
    //esperamos 500 ms
    __delay_ms(500);
    //apagamos el pin RB0 y encendemos el pin RB1
    PORTBbits.RB0 = 0;
    PORTBbits.RB1 = 1;
    //esperamos 500 ms
    __delay_ms(500);
    //apagamos el pin RB1 y encendemos el pin RB2
    PORTBbits.RB1 = 0;

```

```
PORTBbits.RB2 = 1;  
  
//esperamos 500 ms  
__delay_ms(500);  
  
//apagamos el pin RB2  
PORTBbits.RB2 = 0;  
  
}
```

## Link del repositorio

[https://github.com/sag18640/Electronica\\_Digital\\_2.git](https://github.com/sag18640/Electronica_Digital_2.git)