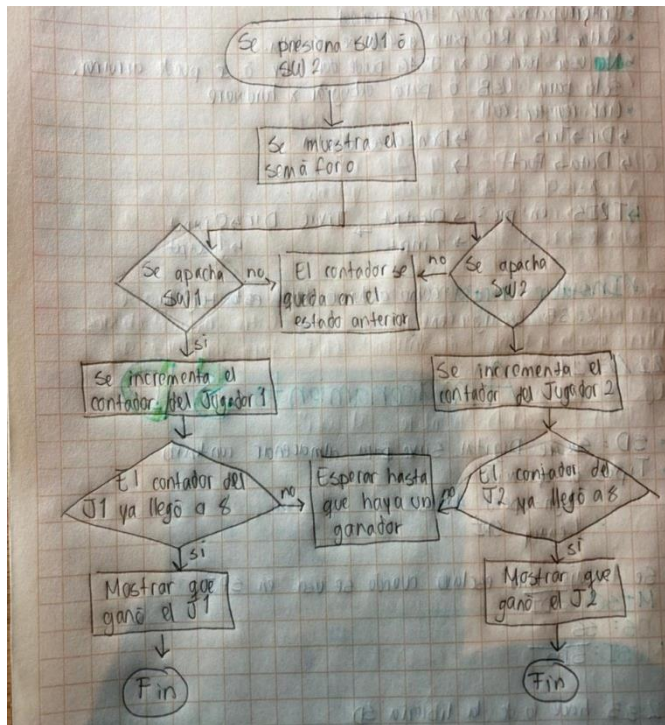


Laboratorio #4

Juego de Carrera Tiva C

Pseudocódigo



Código bien documentado

Lab4 §

```

1 //Nicole Prem 18337
2 //Lab 4: Juego Carrera en Tiva C
3
4 //*****
5 //Variables
6 //*****
7 #define LEDR PF_1 //Led rojo para el semáforo
8 #define LEDV PF_3 //Led verde del semáforo
9 #define LEDA PF_2 //Led azul para mostrar el ganador
10 #define BotonJ1 PUSH1 //Botón del jugador 1
11 #define BotonJ2 PUSH2 //Botón del jugador 2
12 //define RESET PF_RST //Botón de reset
13
14 unsigned char C1; //parámetro de entrada para función de leds J1
15 unsigned char C2; //parámetro de entrada para función de leds J2
16 unsigned char contadorJ1 = 0; //Contador del jugador 1, si llega a 8 gana
17 unsigned char contadorJ2 = 0; //Contador del jugador 2, si llega a 8 gana
18 unsigned int bandera = 1; //bandera para identificar al ganador
19 unsigned int banderaJ1=0; //bandera para identificar si se apachó el botón del J1
20 unsigned int banderaJ2=0; //bandera para identificar si se apachó el botón del J2
21 char debounce=0; //bandera utilizada para implementar el antirrebote
22 int BotonJ1_State=0; //Estado del botón 1
23 int BotonJ2_State=0; //Estado del boton 2
24 int flagJ=0; //bandera para el antirrebote del semáforo la cual cambia dependiendo quién presionó el botón
25
26
27 //*****
28 //Prototipos de funciones
29 //*****
30 void semaforo (void); //función del semáforo
31 void ledsJ1 (unsigned char C1); //función para mostrar los leds del jugador 1
32 void ledsJ2 (unsigned char C2); //función para mostrar los leds del jugador 2
33
34
35 void setup() {
36   pinMode(LEDR, OUTPUT); //Definición del led rojo como output
37   pinMode(LEDV, OUTPUT); //Definición del led verde como output
38   pinMode(BotonJ1, INPUT_PULLUP); //Definición del botón del jugador 1 como input en pullup
39   pinMode(BotonJ2, INPUT_PULLUP); //Definición del botón del jugador 2 como input en pullup
40   // pinMode(RESET, INPUT_PULLUP);
41   //PINES PARA LOS LEDS JUGADOR 1
42   pinMode(PB_5, OUTPUT); //Led0 del jugador 1
43   //pinMode(PB_0, OUTPUT); //Led1 del jugador 1

```

Lab4 §

```

43   pinMode(PB_0, OUTPUT); //Led1 del jugador 1
44   pinMode(PB_1, OUTPUT); //Led2 del jugador 1
45   pinMode(PE_4, OUTPUT); //Led3 del jugador 1
46   pinMode(PE_5, OUTPUT); //Led4 del jugador 1
47   pinMode(PB_4, OUTPUT); //Led5 del jugador 1
48   pinMode(PA_5, OUTPUT); //Led6 del jugador 1
49   pinMode(PA_6, OUTPUT); //Led7 del jugador 1
50   //PINES PARA LOS LEDS JUGADOR 2
51   pinMode(PD_0, OUTPUT); //Led0 del jugador 2
52   pinMode(PD_1, OUTPUT); //Led1 del jugador 2
53   pinMode(PD_2, OUTPUT); //Led2 del jugador 2
54   pinMode(PD_3, OUTPUT); //Led3 del jugador 2
55   pinMode(PE_1, OUTPUT); //Led4 del jugador 2
56   pinMode(PE_2, OUTPUT); //Led5 del jugador 2
57   pinMode(PE_3, OUTPUT); //Led6 del jugador 2
58   pinMode(PD_6, OUTPUT); //Led7 del jugador 2
59 }
60
61 void loop() {
62   //Condiciones iniciales:
63   contadorJ1 = 0;
64   contadorJ2 = 0;
65   ledsJ1(contadorJ1);
66   ledsJ2(contadorJ2);
67   banderaJ1=0;
68   banderaJ2=0;
69   BotonJ1_State=digitalRead(BotonJ1);
70   BotonJ2_State=digitalRead(BotonJ2);
71   debounce=0;
72   flagJ=0;
73   //Determinación de quién presionó el botón de inicio junto con antirrebote del mismo botón
74   if (BotonJ1_State==LOW or BotonJ2_State==LOW){
75     if (BotonJ1_State==LOW){
76       flagJ=2;
77     }
78     else if (BotonJ2_State==LOW){
79       flagJ=1;
80     }
81     while (debounce==0){ //antirrebote
82       BotonJ1_State=digitalRead(BotonJ1);
83       BotonJ2_State=digitalRead(BotonJ2);
84       if (BotonJ1_State==HIGH and flagJ==2){
85         debounce=1;

```

Lab4 §

```

85     debounce=1;
86   }
87   else if(BotonJ2_State==HIGH and flagJ==1){
88     debounce=1;
89   }
90 }
91
92 semaforo();
93 bandera=1;
94
95 while (bandera == 1) {
96   BotonJ1_State=digitalRead(BotonJ1);
97   BotonJ2_State=digitalRead(BotonJ2);
98   debounce=0;
99   if (BotonJ1_State == LOW) { //Se presiona botón J1
100     while(debounce==0){//Antirrebote
101       BotonJ1_State=digitalRead(BotonJ1);
102       BotonJ2_State=digitalRead(BotonJ2);
103       if(BotonJ1_State==HIGH){
104         debounce=1;
105       }
106     }
107     delay(200);
108     contadorJ1++; //se incrementa el contador
109     ledsJ1(contadorJ1); //se va a la función ledsJ1 para encender los leds del jugador 1
110
111     if (contadorJ1 == 8) { //verificación si ya ganó
112       bandera = 2; //se sale del ciclo
113     }
114   }
115   else if (BotonJ2_State == LOW) { //Se presiona botón J2
116     while(debounce==0){//Antirrebote
117       BotonJ1_State=digitalRead(BotonJ1);
118       BotonJ2_State=digitalRead(BotonJ2);
119       if(BotonJ2_State==HIGH){
120         debounce=1;
121       }
122     }
123     delay(200);
124     contadorJ2++; //se incrementa el contador
125     ledsJ2(contadorJ2); //se va a la función ledsJ2 para encender los leds del jugador 2
126
127     if (contadorJ2 == 8) { //verificación si ya ganó

```

Lab4 §

```

127     if (contadorJ2 == 8) { //verificación si ya ganó
128       bandera = 2; //se sale del ciclo
129     }
130   }
131 }
132 }
133 }
134 }
135 }
136 }
137 }
138 }
139 }
140
141 //*****
142 //Funciones
143 //*****
144 //Función del semáforo
145 void semaforo (void){
146   digitalWrite(LED_R, HIGH); //Encender led rojo
147   delay(1000); //esperamos 1 segundo
148   digitalWrite(LED_V, HIGH); //Encender led rojo y verde al mismo tiempo para crar color amarillo
149   delay(1000); //esperamos 1 segundo
150   digitalWrite(LED_R, LOW);
151   delay(1000);
152   digitalWrite(LED_V, LOW);
153   delay(1000);
154 }
155
156
157 // Rutina para desplegar los leds del J1
158 void ledsJ1(unsigned char C1) {
159   if (C1 == 8) { //se verifica si ya presionó el botón 8 veces y si ya ganó entonces se enciende el led azul tres veces
160     digitalWrite(PA_5, LOW); //se apaga el led 6
161     digitalWrite(PA_6, HIGH); //se enciende led 7
162     digitalWrite(LEDV, HIGH);
163     delay(1000);
164     digitalWrite(LEDV, LOW);
165     delay(1000);
166     digitalWrite(LEDV, HIGH);
167     delay(1000);
168     digitalWrite(LEDV, LOW);
169     delay(1000);

```

Lab4 §

```

169     delay(1000);
170     digitalWrite(LEDV, HIGH);
171     delay(1000);
172     digitalWrite(LEDV, LOW);
173     delay(1000);
174     digitalWrite(PA_6, LOW);
175 }
176 /*Case que indica el led que se enciende dependiendo de la car
177 que ha presionado el botón */
178 switch (C1) {
179     case 0:
180         digitalWrite(PB_5, LOW); //Led0 del jugador 1
181         digitalWrite(PB_0, LOW); //Led1 del jugador 1
182         digitalWrite(PB_1, LOW); //Led2 del jugador 1
183         digitalWrite(PE_4, LOW); //Led3 del jugador 1
184         digitalWrite(PE_5, LOW); //Led4 del jugador 1
185         digitalWrite(PB_4, LOW); //Led5 del jugador 1
186         digitalWrite(PA_5, LOW); //Led6 del jugador 1
187         digitalWrite(PA_6, LOW); //Led7 del jugador 1
188         break;
189     case 1: //se enciende led 0
190         digitalWrite(PB_5, HIGH);
191         break;
192     case 2: //se enciende led 1
193         digitalWrite(PB_5, LOW);
194         digitalWrite(PB_0, HIGH);
195         break;
196     case 3: //se enciende led 2
197         digitalWrite(PB_0, LOW);
198         digitalWrite(PB_1, HIGH);
199         break;
200     case 4: //se enciende led 3
201         digitalWrite(PB_1, LOW);
202         digitalWrite(PE_4, HIGH);
203         break;
204     case 5: //se enciende led 4
205         digitalWrite(PE_4, LOW);
206         digitalWrite(PE_5, HIGH);
207         break;
208     case 6: //se enciende led 5
209         digitalWrite(PE_5, LOW);
210         digitalWrite(PB_4, HIGH);
211         break;

```

Lab4 §

```

211         break;
212     case 7: //se enciende led 6
213         digitalWrite(PB_4, LOW);
214         digitalWrite(PA_5, HIGH);
215         break;
216 }
217 }
218 }
219
220 // Rutina para desplegar los leds del J2
221 void ledsJ2(unsigned char C2) {
222     if (C2 == 8) { //se verifica si ya presionó 8 veces el botón y si ya ganó se enciende el led rojo tres veces para mostrar que el es el ganador
223         digitalWrite(PE_3, LOW); //se apaga el led 6
224         digitalWrite(PD_6, HIGH); //se enciende led 7
225         digitalWrite(LED7, HIGH);
226         delay(1000);
227         digitalWrite(LED7, LOW);
228         delay(1000);
229         digitalWrite(LED7, HIGH);
230         delay(1000);
231         digitalWrite(LED7, LOW);
232         delay(1000);
233         digitalWrite(LED7, HIGH);
234         delay(1000);
235         digitalWrite(LED7, LOW);
236         delay(1000);
237         digitalWrite(PD_6, LOW);
238     }
239     /*Case que indica el led que se enciende dependiendo de la cantidad de veces
240 que ha presionado el botón */
241     switch (C2) {
242         case 0:
243             digitalWrite(PD_0, LOW); //Led0 del jugador 2
244             digitalWrite(PD_1, LOW); //Led1 del jugador 2
245             digitalWrite(PD_2, LOW); //Led2 del jugador 2
246             digitalWrite(PD_3, LOW); //Led3 del jugador 2
247             digitalWrite(PE_1, LOW); //Led4 del jugador 2
248             digitalWrite(PE_2, LOW); //Led5 del jugador 2
249             digitalWrite(PE_3, LOW); //Led6 del jugador 2
250             digitalWrite(PD_6, LOW); //Led7 del jugador 2
251             break;
252         case 1: //se enciende led 0
253             digitalWrite(PD_0, HIGH);

```

```

249     digitalWrite(PE_3, LOW); //Led6 del jugador 2
250     digitalWrite(PD_6, LOW); //Led7 del jugador 2
251     break;
252     case 1: //se enciende led 0
253         digitalWrite(PD_0, HIGH);
254         break;
255     case 2: //se enciende led 1
256         digitalWrite(PD_0, LOW);
257         digitalWrite(PD_1, HIGH);
258         break;
259     case 3: //se enciende led 2
260         digitalWrite(PD_1, LOW);
261         digitalWrite(PD_2, HIGH);
262         break;
263     case 4: //se enciende led 3
264         digitalWrite(PD_2, LOW);
265         digitalWrite(PD_3, HIGH);
266         break;
267     case 5: //se enciende led 4
268         digitalWrite(PD_3, LOW);
269         digitalWrite(PE_1, HIGH);
270         break;
271     case 6: //se enciende led 5
272         digitalWrite(PE_1, LOW);
273         digitalWrite(PE_2, HIGH);
274         break;
275     case 7: //se enciende led 6
276         digitalWrite(PE_2, LOW);
277         digitalWrite(PE_3, HIGH);
278         break;
279     }
280 }

```

Link de Github

<https://github.com/nicoleprem/Digital-2.git>