```
#include "mbed.h"
#include "pinout.h"
#include "to_7seg.h"

// seven segment display anodes
// when in a int8_t, they are 0b-GFEDCBA
BusOut      g_seven_seg(SGA_PIN, SGB_PIN, SGC_PIN, SGD_PIN,
                        SGE_PIN, SGF_PIN, SGG_PIN);

// display cathodes
DigitalOut  g_dsr(DSR_PIN);
DigitalOut  g_dsl(DSL_PIN);

// leds
BusOut      g_leds(LDR_PIN, LDM_PIN, LDL_PIN);


//ticker
static Ticker tick_100ms;

//evento
static bool volatile tick_100ms_evnt;
//interrupcion
static void tick_100ms_isr(void){
  tick_100ms_evnt = true;
}

//interrupcion
static InterruptIn swm (SWM_PIN);

//eventos botones
static bool volatile swm_evnt;

//interrupciones botones
static void swm_isr(void){
  swm_evnt = true;
}

//REBOTES
static Timeout t_4ms;
static bool volatile t_4ms_evnt;
static void t_4ms_isr (void){
  t_4ms_evnt = true;
}

static int8_t g_cnt_sw = 0;    //número de veces que se acciona el pulsador central

int main (void) {

 bool gb_led_seq_on = true;

  g_leds = 4;

  tick_100ms.attach_us(tick_100ms_isr,100000);

  //pulsos
  swm.mode(PullUp);
  swm.fall(swm_isr);

  for (;;) {

    if(tick_100ms_evnt){
   tick_100ms_evnt = false;

      if(gb_led_seq_on){
      g_leds = (g_leds==1) ? 4 : (g_leds >> 1);

      }else{
        g_leds = g_leds;
      }
    }

    if(swm_evnt){
      swm_evnt = false;
      t_4ms.attach_us(t_4ms_isr,4000);
    }

    if (t_4ms_evnt){
      t_4ms_evnt = false;

      if(swm ==0){
          gb_led_seq_on = !gb_led_seq_on;
      }

```

```
85        }
86
87        __disable_irq();
88        if(!tick_100ms_evnt && !swm_evnt && !t_4ms_evnt){
89           __WFI();
90        }
91        __enable_irq();
92
93        // [2] segundo bloque de código que puede rellenar ++++++++++++++++++
94
95        //Parte de los leds preguntada
96
97
98        // [2] fin del segundo bloque de código ----------------------------
99
100     } // for (;;)
101    } // main()
102
```