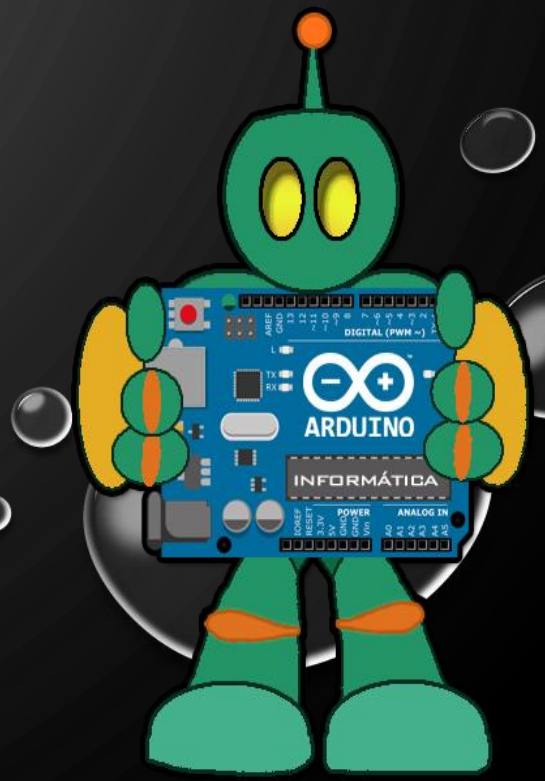
COMUNIDAD ARDUINO OPEN SOURCE

CLASE 5 – INTERRUPCIONES ARDUINO III



INTERRUPCIONES

Una interrupción es un proceso que permite reaccionar de forma rápida a una petición, impidiendo así la continuidad de otro proceso generado.

Existen dos tipos de interrupciones:

- Interrupciones por software

(Timers)

- Interrupciones por hardware

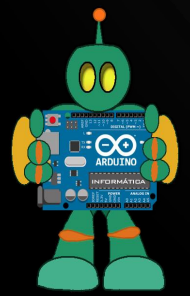
(Externas)

TUTOR: NAGIB LUIS VALLEJOS M.

PINOUT

Las placas arduino o propiamente los microcontroladores **Atmel** tienen pines específicos que nos sirven para generar interrupciones, estos pines son:

PLACA	INT0	INT1	INT2	INT3	INT4	INT5
UNO	2	3				
NANO	2	3				
PRO MINI	2	3				
MEGA	2	3	21	20	19	18
LEONARDO	3	2	0	1	7	
DUE	# interrupciones = # pines					
DUE = UNO WIFI, ZERO, MKR FAMILY Y 101						



TUTOR: NAGIB LUIS VALLEJOS M.

ATTACHINTERRUPT()

Permite definir un pin como puerto de interrupción, esta función está compuesta por 3 parámetros (pin, ISR, modo) y maneja la siguiente sintaxis:

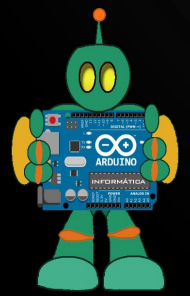
```
attachInterrupt(pin,ISR,mode);
```

- pin: Es el numero ordinal del pin digital donde se encuentra la interrupción. *Ordinal 0 -> D2, Ordinal 1 -> D3*. Para facilitar facilitar su uso, arduino nos permite usar la función digitalPinInterrupt(pin).

```
digitalPinInterrupt(2);
```

- ISR(Interruption Service Rutine): Es un método que se ejecuta cuando se produce la interrupción.

Es un método que no admite parámetros y no devuelve ningún valor, por tanto, trabaja solo con estructuras void().

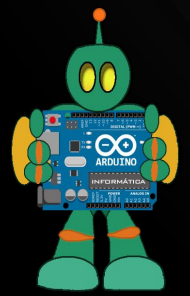


TUTOR: NAGIB LUIS VALLEJOS M.

ATTACHINTERRUPT()

Para realizar el uso correcto de la función ***ISR***

- El segmento de código debe tener el menor tiempo de ejecución posible.
- Nos sirve para modificar una variable o incrementar un contador.
- No se puede utilizar la función *millis(), delay() y micros()* salvo que el tiempo no supere los 500µs.
- La función *delayMicroseconds()* funcionará en el rango de los 500µs.
- Para poder modificar una variable externa a la ISR, debe ser declarada como ***volatile***, un indicador volatile indica al compilador que la variable tiene que ser consultada antes de ser usada, dado que puede haber sido modificada de forma ajena al flujo normal del programa.



TUTOR: NAGIB LUIS VALLEJOS M.

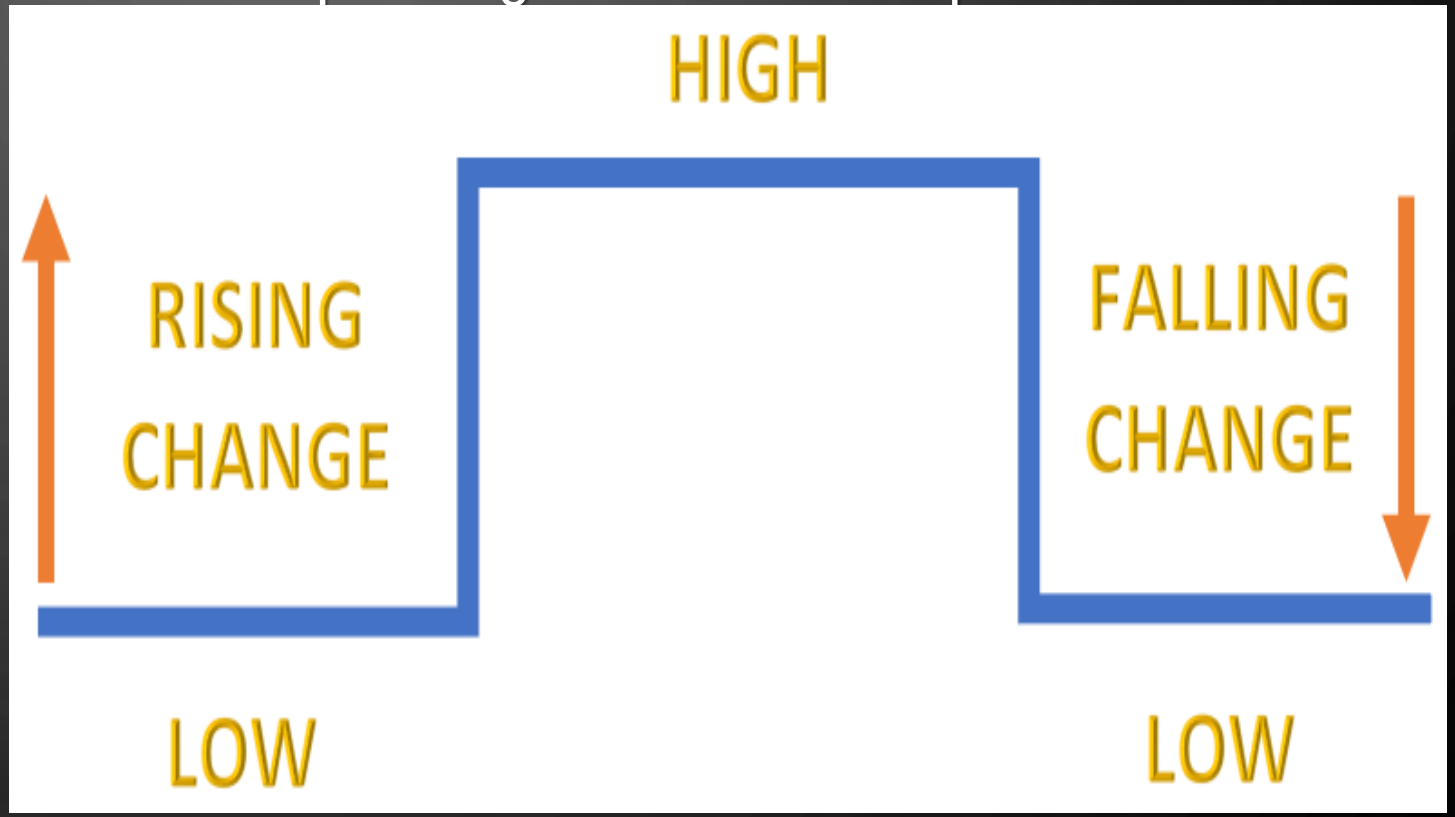
ATTACHINTERRUPT()

- **Mode:** existen 4 modos en los cuales se puede generar la interrupción:

LOW: La interrupción se dispara cuando el pin es 0.

RISING: Se dispara en el franco de subida (cuando pasa de LOW a HIGH).

FALLING: Se dispara en el franco de bajada (cuando pasa de HIGH a LOW).



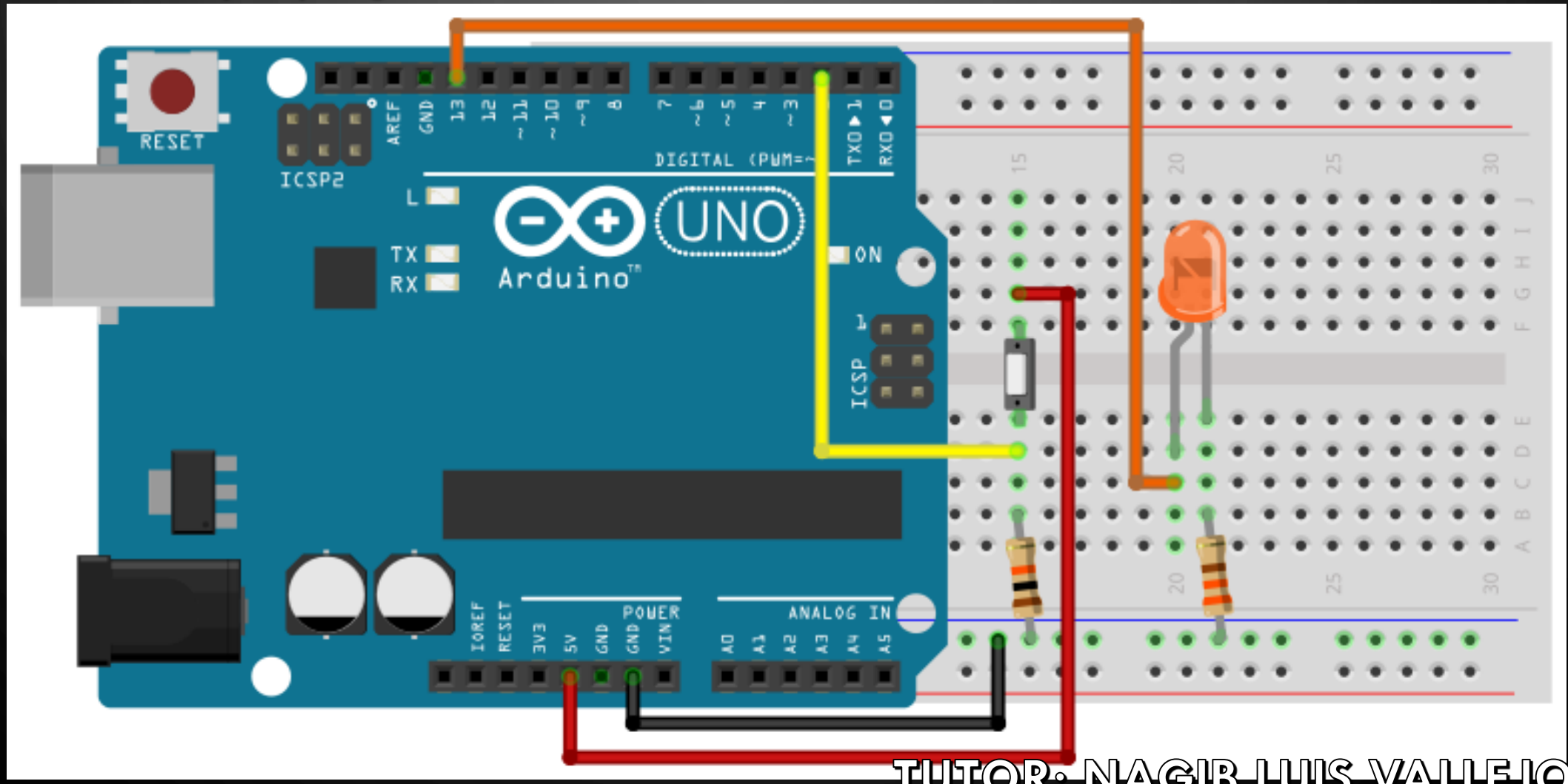
CHANGE: Se dispara cuando pase de HIGH a LOW o viceversa.

HIGH: Se dispara cuando el pin está en HIGH. *(Due, zero, mkr1000)*

TUTOR: NAGIB LUIS VALLEJOS M.

EJERCICIO 1 - CIRCUITO

Encender y apagar un LED a través del modo FALLING



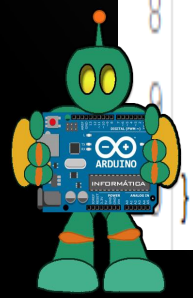
TUTOR: NAGIB LUIS VALLEJOS M.

EJERCICIO 1 - SOLUCIÓN

Encender y apagar un LED a través del modo FALLING

S5-E1

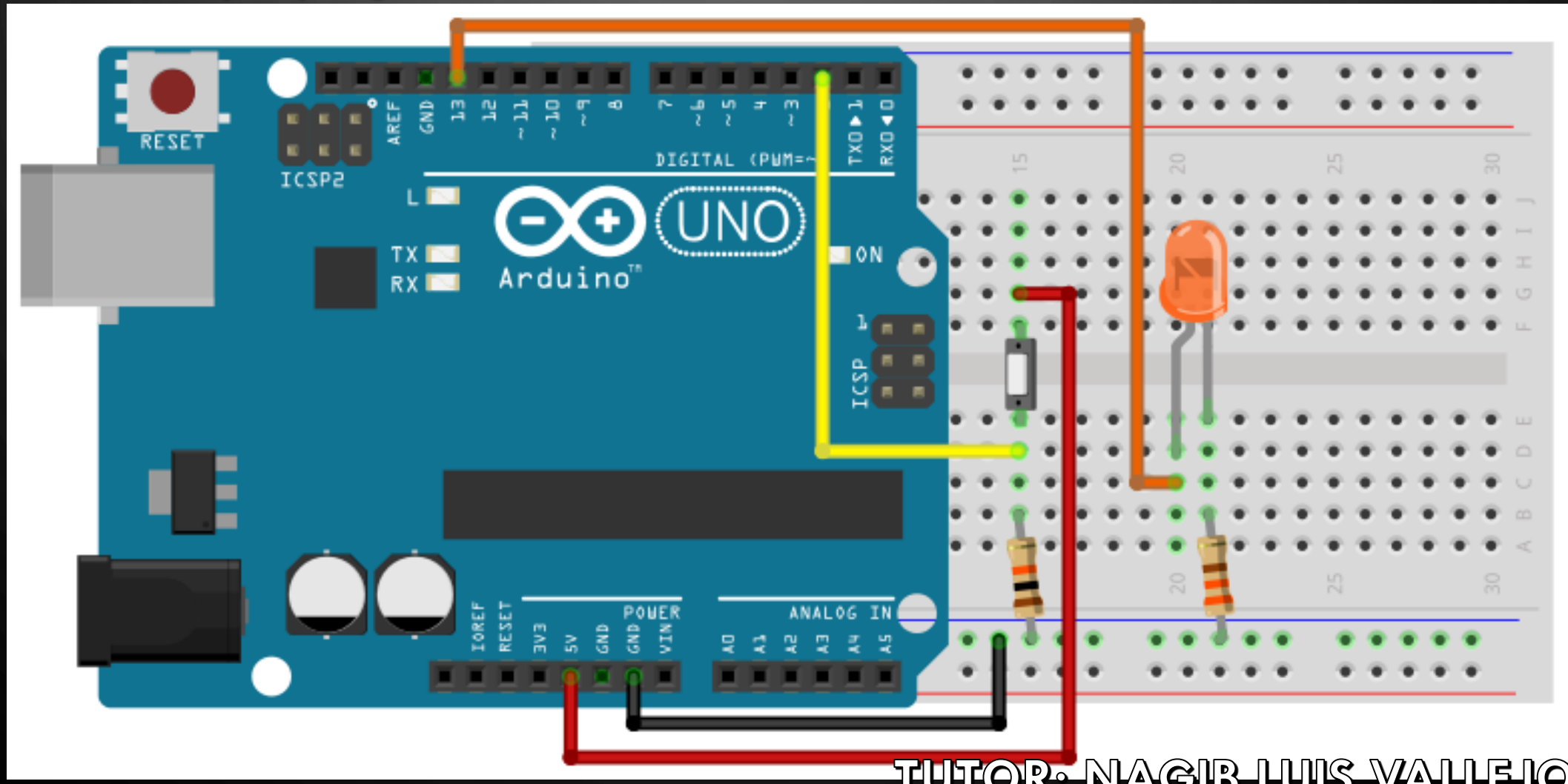
```
int pulsador=2,led=13;
boolean estado=false;
long tiempo=0;
void setup() {
  pinMode(pulsador,INPUT);
  pinMode(led,OUTPUT);
  //attachInterrupt(pin,isr,mode);
  attachInterrupt(digitalPinToInterrupt(pulsador),cambio,FALLING);
  Serial.begin(9600);
}
void loop() {
  delay(2000);
}
void cambio(){
  if(millis()>(tiempo+20)){
    estado=!estado;
    digitalWrite(led,estado);
    Serial.println("Cambio de estado: "+String(estado));
  }
  tiempo=millis();
}
```



TUTOR: NAGIB LUIS VALLEJOS M.

EJERCICIO 2 - CIRCUITO

Encender y apagar un LED a través del modo CHANGE

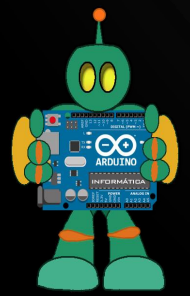


TUTOR: NAGIB LUIS VALLEJOS M.

EJERCICIO 2 - SOLUCIÓN

S5-E2

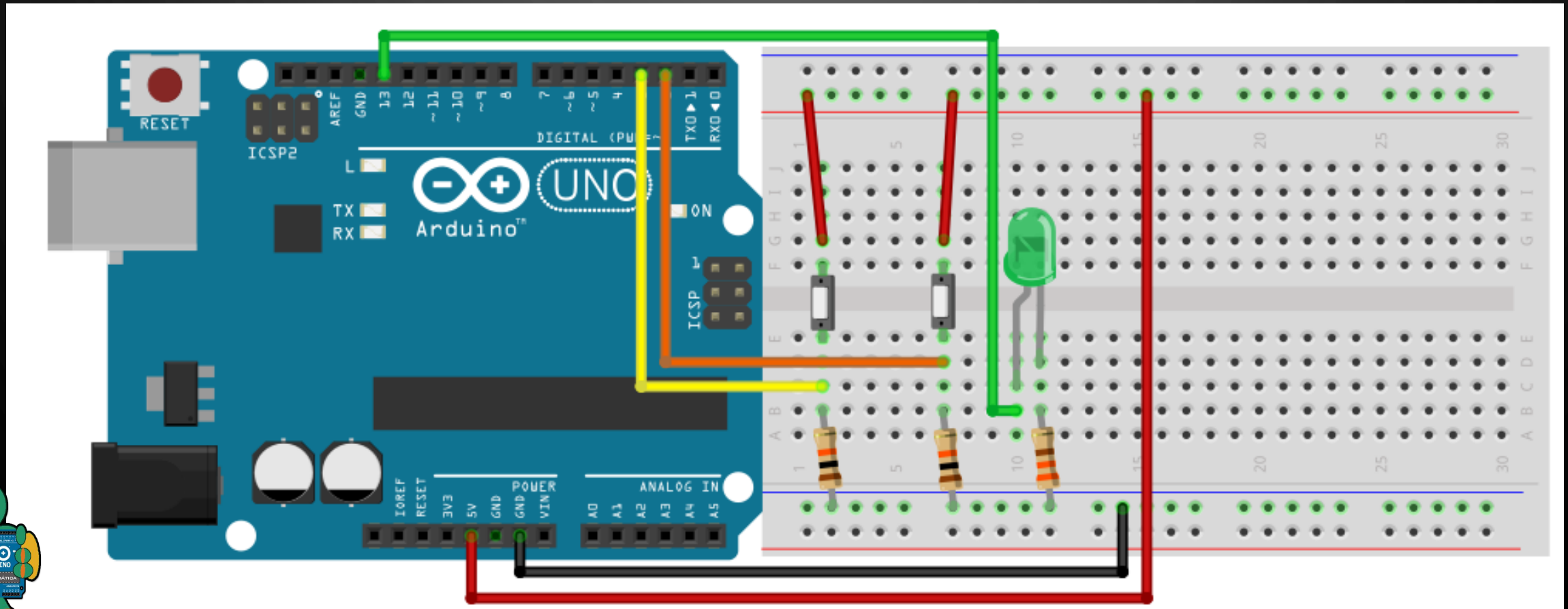
```
1  int pulsador=2,led=13;
2  volatile byte estado=0;
3  void setup() {
4    pinMode(pulsador,INPUT);
5    pinMode(led,OUTPUT);
6    attachInterrupt(digitalPinToInterrupt(pulsador),parpadeo,CHANGE);
7    Serial.begin(9600);
8  }
9
10 void loop() {
11   digitalWrite(led,estado);
12   Serial.println("Cambio de estado: "+String(estado));
13 }
14
15 void parpadeo(){
16   estado=!estado;
17 }
```



TUTOR: NAGIB LUIS VALLEJOS M.

EJERCICIO 3 - CIRCUITO

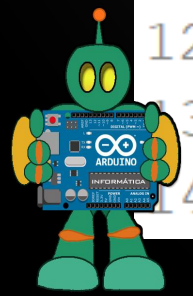
Encender y apagar 1 LED con 2 pulsadores a través del modo FALLING



TUTOR: NAGIB LUIS VALLEJOS M.

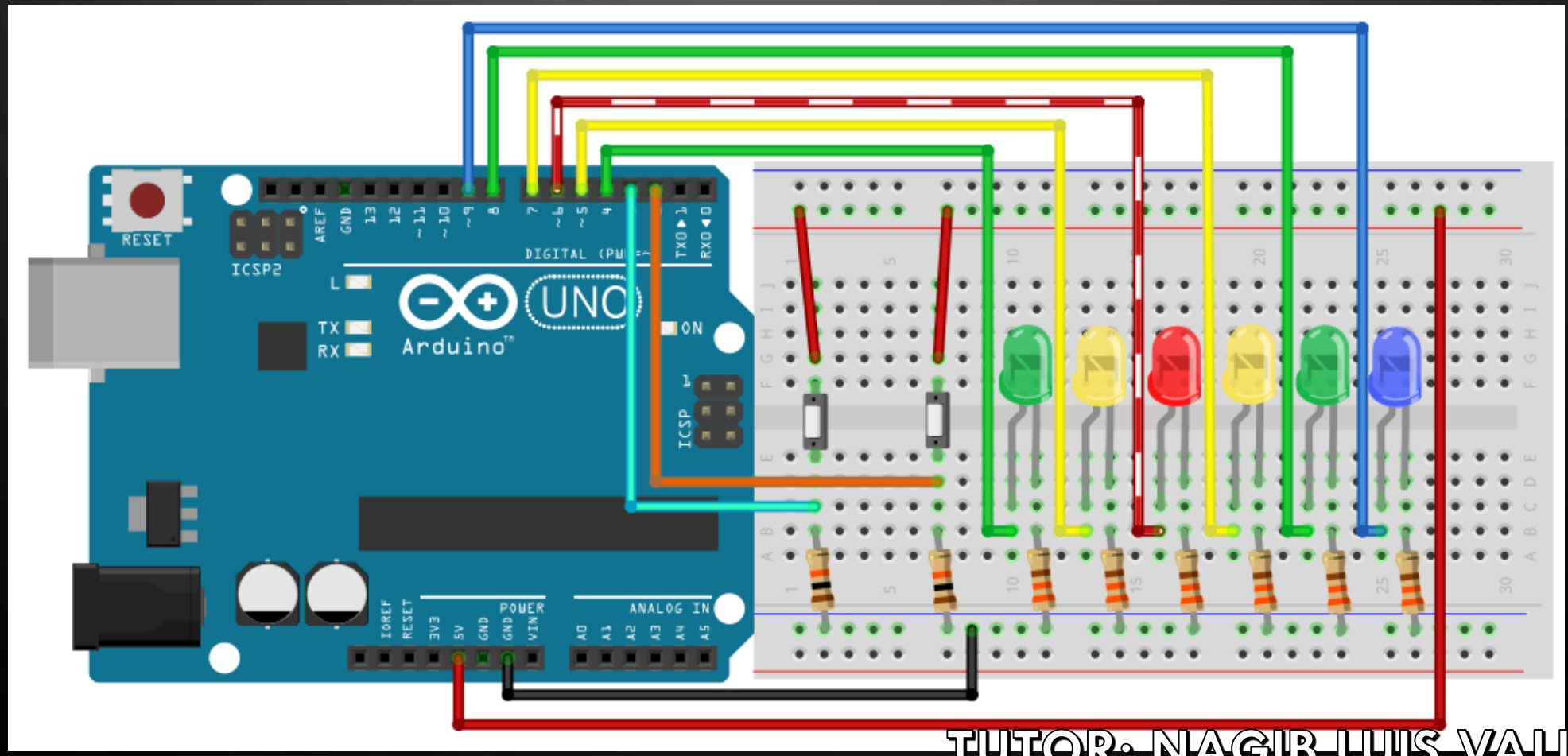
EJERCICIO 3 - SOLUCIÓN

```
1  int p1=3,p2=2,led=13;
2  boolean estado=false;
3  long tiempo=0;
4  void setup() {
5    pinMode(p1,INPUT);
6    pinMode(p2,INPUT);
7    pinMode(led,OUTPUT);
8    attachInterrupt(digitalPinToInterrupt(p1),encender,FALLING);
9    attachInterrupt(digitalPinToInterrupt(p2),apagar,FALLING);
10   Serial.begin(9600);
11 }
12 void loop() {
13   delay(2000);
14 }
```



EJERCICIO 4 - CIRCUITO

Variar la velocidad de encendido/apagado del juego de luces través del modo RISING

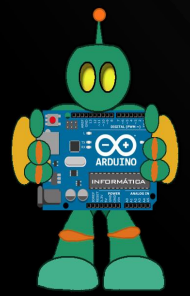


TUTOR: NAGIB LUIS VALLEJOS M.

EJERCICIO 4 - SOLUCIÓN

S5-E4

```
1  volatile int velocidad=20;
2  int maximo=1000,minimo=20,aumento=20;
3  int leds [6]={4,5,6,7,8,9};
4  //              0 1 2 3 4 5
5  void setup(){
6    for(int i=0;i<6;i++){
7        pinMode(leds[i],OUTPUT);
8    }
9    attachInterrupt(digitalPinToInterrupt(2),lento,RISING);
10   attachInterrupt(digitalPinToInterrupt(3),rapido,RISING);
11   Serial.begin(9600);
12   velocidad=minimo;
13 }
14
15 void loop()  {
16   for(int i=0;i<6;i++){
17     if(i>0){
18       digitalWrite(leds[i-1],0);
19     }
20     digitalWrite(leds[i],1);
21     delay(velocidad);
22   }
23   digitalWrite(leds[5],0);
```

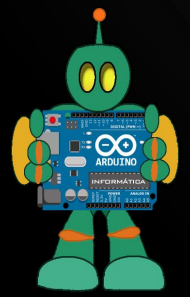


TUTOR: NAGIB LUIS VALLEJOS M.

EJERCICIO 4 - SOLUCIÓN

S5-E4

```
24    for(int i=5;i>=0;i--){
25      if(i<5){
26        digitalWrite(leds[i+1],0);
27      }
28      digitalWrite(leds[i],1);
29      delay(velocidad);
30    }
31    digitalWrite(leds[0],0);
32    Serial.println("velocidad: "+String(velocidad));
33  }
34
35  void lento(){
36    velocidad=velocidad - aumento;
37    if(velocidad<minimo){
38      velocidad=minimo;
39    }
40  }
41  void rapido(){
42    velocidad=velocidad + aumento;
43    if(velocidad>maximo){
44      velocidad=maximo;
45    }
46  }
```



TUTOR: NAGIB LUIS VALLEJOS M.

CONTACTOS

(+591) 63096640

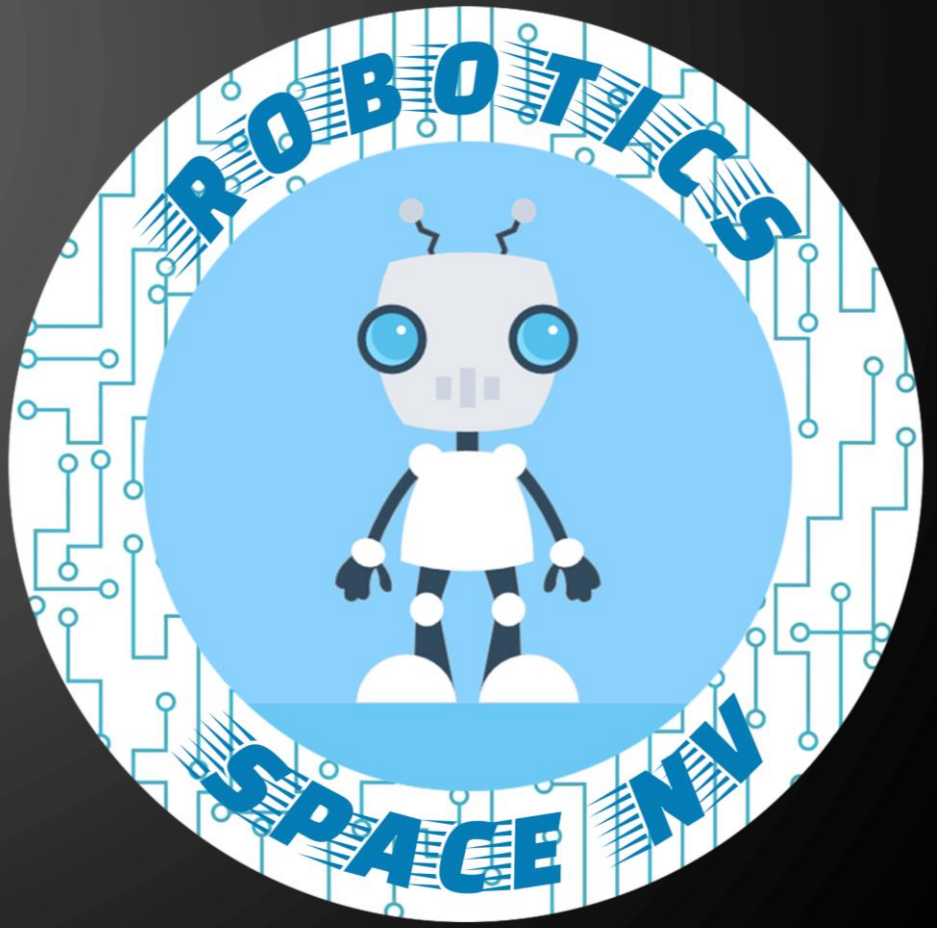
robotics.space.nv@gmail.com

fb.me/RoboticsSpaceNV

@NagibVallejos

Robotics Space NV

https://github.com/nagibvalejos/Robotics-Space-NV



TUTOR: NAGIB LUIS VALLEJOS M.