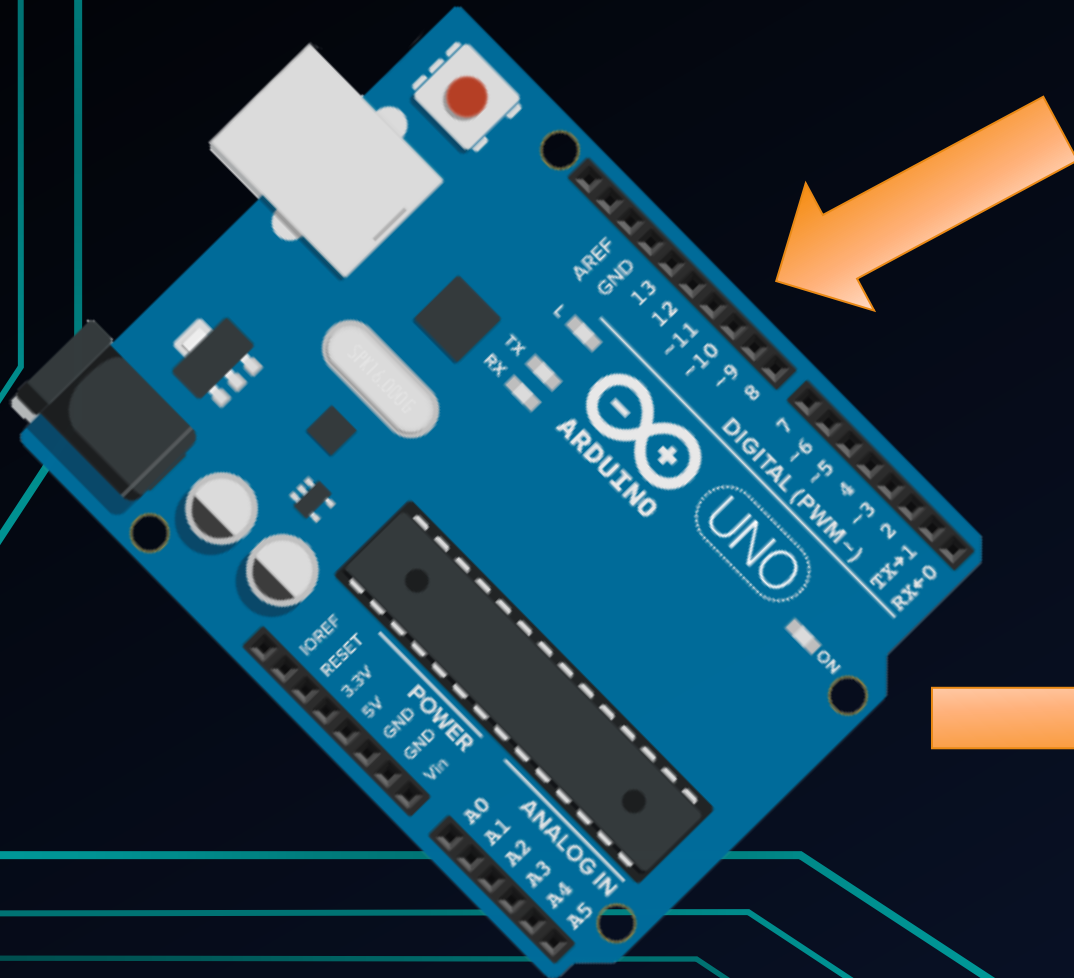
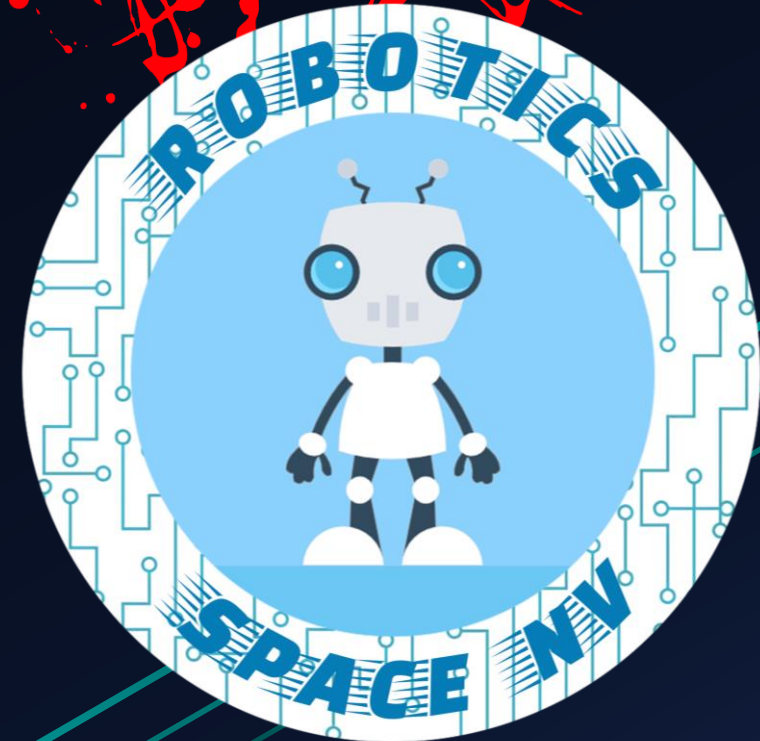


# ENTRADAS DIGITALES

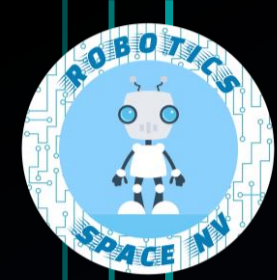
## CLASE 3



SUSCRÍBETE



TUTOR: NAGIB LUIS VALLEJOS M.



# ENTRADAS DIGITALES

Una **entrada digital** es aquella que permite enviar una señal del ambiente hacia arduino.

Los valores que se pueden generar son:

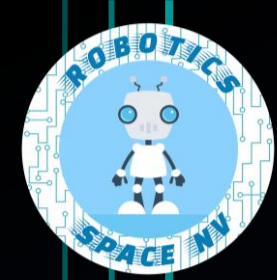
**0=LOW=Apagado**  
**1=HIGH=Encendido**

Para declararla se utiliza la siguiente sintaxis:

**`pinMode(pin,INPUT);`**



**TUTOR: NAGIB LUIS VALLEJOS M.**



# DATO

## ¿Qué es un dato?

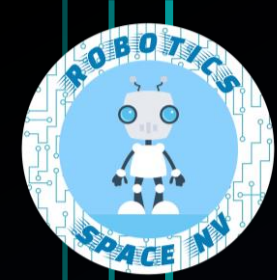
Es una representación simbólica de un atributo o variable cualitativa o cuantitativa. Describen hechos empíricos, sucesos y entidades.

## ¿Qué tipos de datos existen en arduino?

Los datos en arduino se clasifican en:

**int, double, float, long, char,  
bool, byte, string**



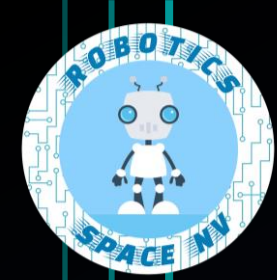


# TIPOS DE DATOS

TIPO	TAMAÑO	RANGO
bool	1 byte	0 – 1 (True o False)
byte / unsigned char	1 byte	0 – 255
char	1 byte	-128 – 127
int	2 bytes	-32768 – 32767
unsigned int	2 bytes	0 – 65535
long	2 bytes	-2147483648 – 2147483647
unsigned long	4 bytes	0 – 4294967295
float / double	4 bytes	- 3.4028235E38 – 3.4028235E38
string	1 byte + x	Array de caracteres

TUTOR: NAGIB LUIS VALLEJOS M.





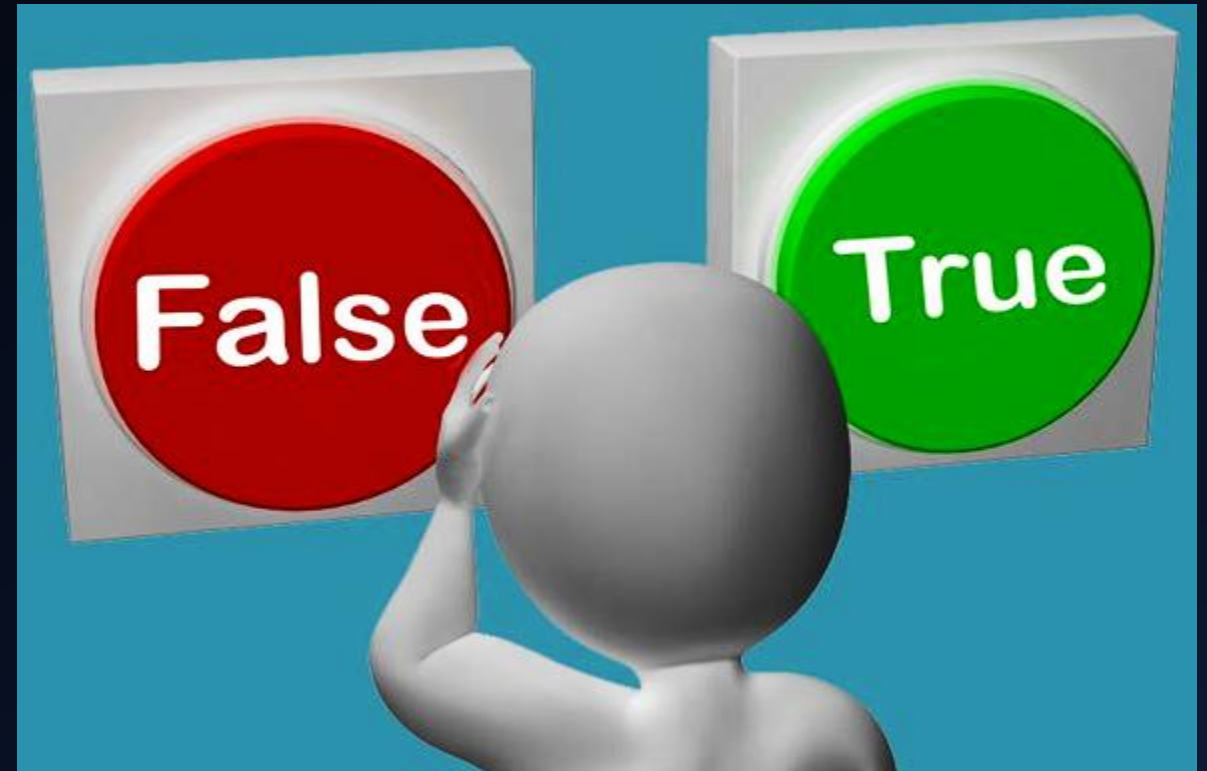
# ESTRUCTURAS DE CONTROL

Una estructura de control permite modificar el flujo de un programa y se puede dividir en las siguientes: If, case, while, do while y for

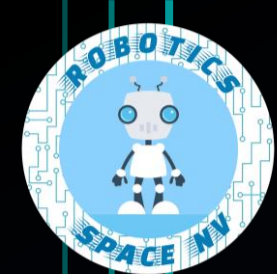
## If - Condicional

Permite indicar la veracidad de una condición, por lo cual emplea operadores comparativos los cuales son:

**>, <, >=, <=, ==, !=**



**TUTOR: NAGIB LUIS VALLEJOS M.**



# OPERADORES COMPARATIVOS

22 == 30 *F*

12 != 30 *V*

10 > 15 *F*

15 < 30 *V*

9 >= 2 *V*

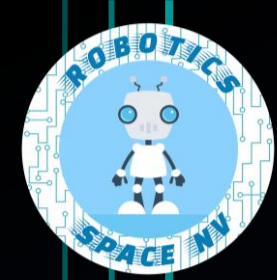
6 <= 6 *V*

Donde:

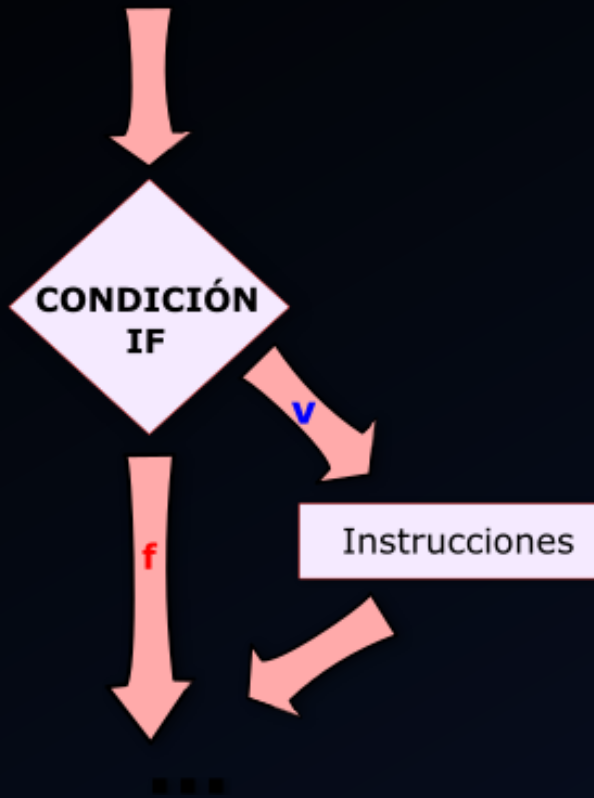
V = 1 o HIGH.

F = 0 o LOW.

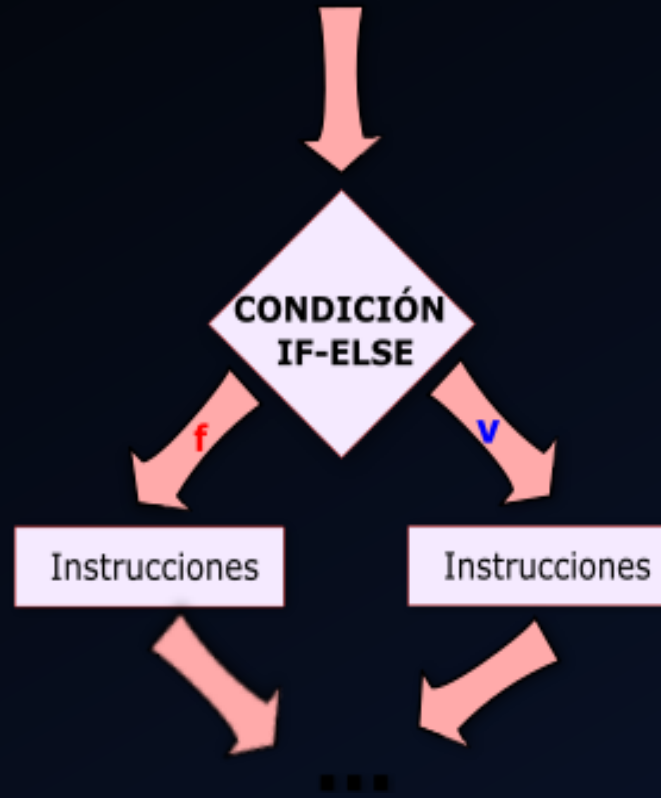
TUTOR: NAGIB LUIS VALLEJOS M.



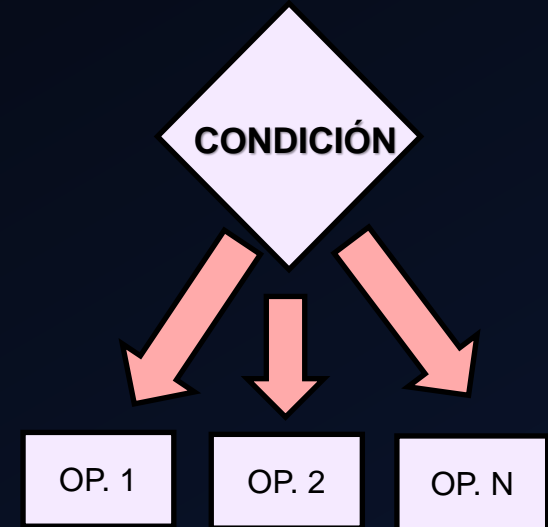
# TIPOS DE CONDICIONALES



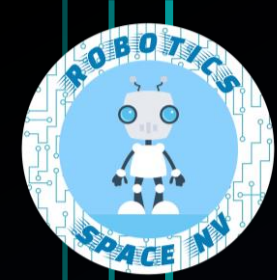
**CONDICIONAL  
SIMPLE**



**CONDICIONAL  
DOBLE**



**CONDICIONAL  
MÚLTIPLE**



# PULSADOR

Es un interruptor eléctrico que permite activar una función al ser presionado

Cuenta con dos tipos de contactos: NA – NC.

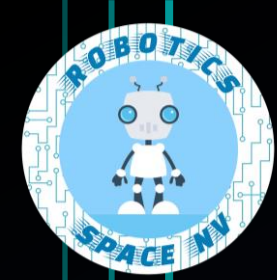
Funciona como un Switch permitiendo el cambio de estado a través de una entrada digital y trabaja con una resistencia de  $10\text{ K}\Omega$ .

## TIPOS DE PULSADORES



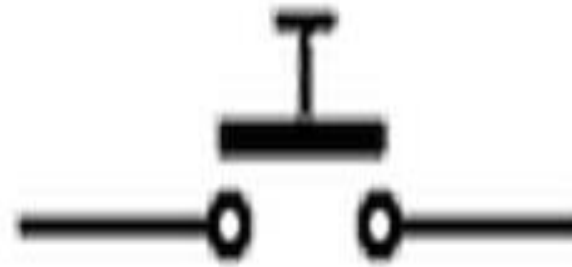
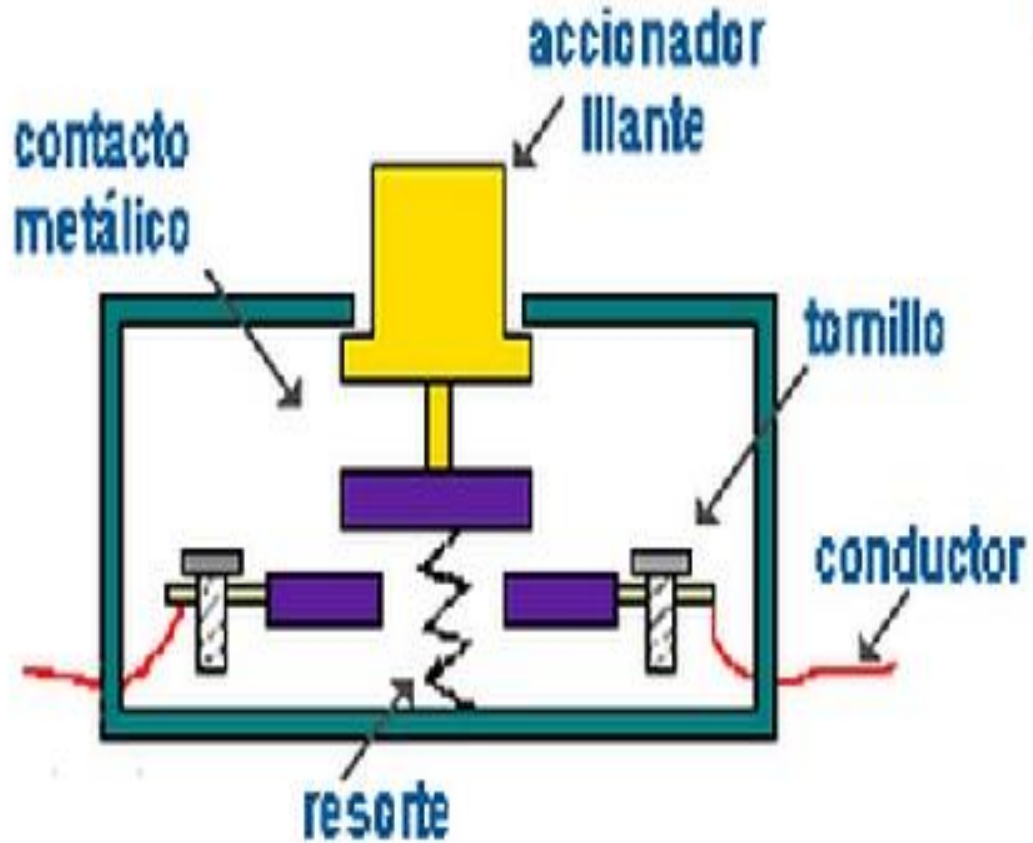
TUTOR: NAGIB LUIS VALLEJOS M.





# COMPOSICIÓN DE UN PULSADOR

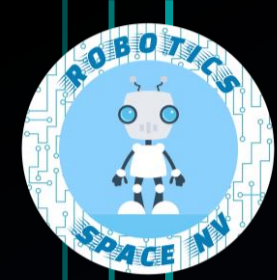
## Pulsadores



Pulsador NA



Pulsador NC



# VARIABLES Y CONSTANTES

Una constante es un tipo de dato que siempre tendrá el mismo valor a lo largo de tiempo, por ejemplo:

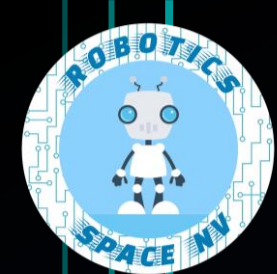
**CI, pi, días de la semana, país de nacimiento**

Una variable es un tipo de dato que cambia a lo largo de la ejecución de un programa, por ejemplo:

**Edad, peso, cantidad de dinero, lugar de residencia**

En arduino el uso de variables y constantes permiten dar un mayor entendimiento a nuestros programas ya que nos permiten identificar los componentes de uso y pines donde se encuentran conectados, sin necesidad de tener que ver el circuito de manera física.

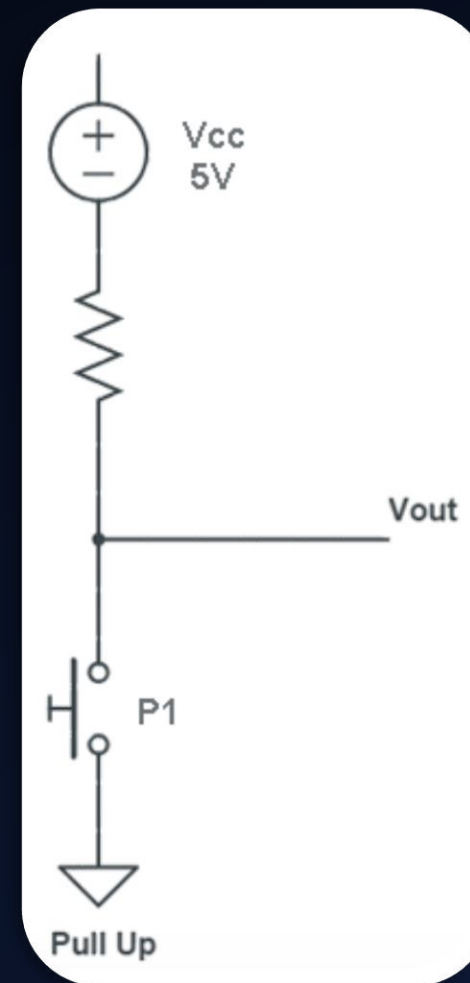
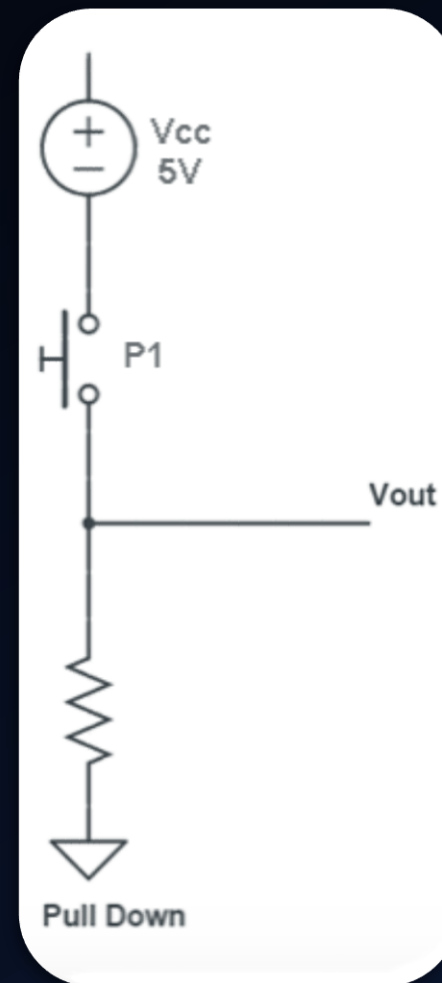
**TUTOR: NAGIB LUIS VALLEJOS M.**



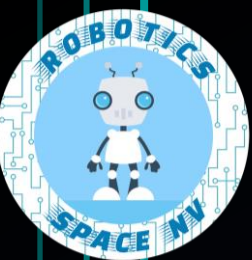
# FUNCIÓN INPUT E INPUT\_PULLUP

La función **INPUT** permite habilitar un pin como entrada y requiere del uso de una resistencia para realizar una correcta lectura pin de entrada, resistencia **pull down** hace referencia al tipo de conexión que se realiza a través en nuestro circuito.

La función **INPUT\_PULLUP** permite habilitar un pin como entrada digital y activa la resistencia interna con la que cuentan las placas arduino, a través de esta ya no es necesario conectar una resistencia al pin de entrada, pero a su vez trabaja con ***lógica inversa***

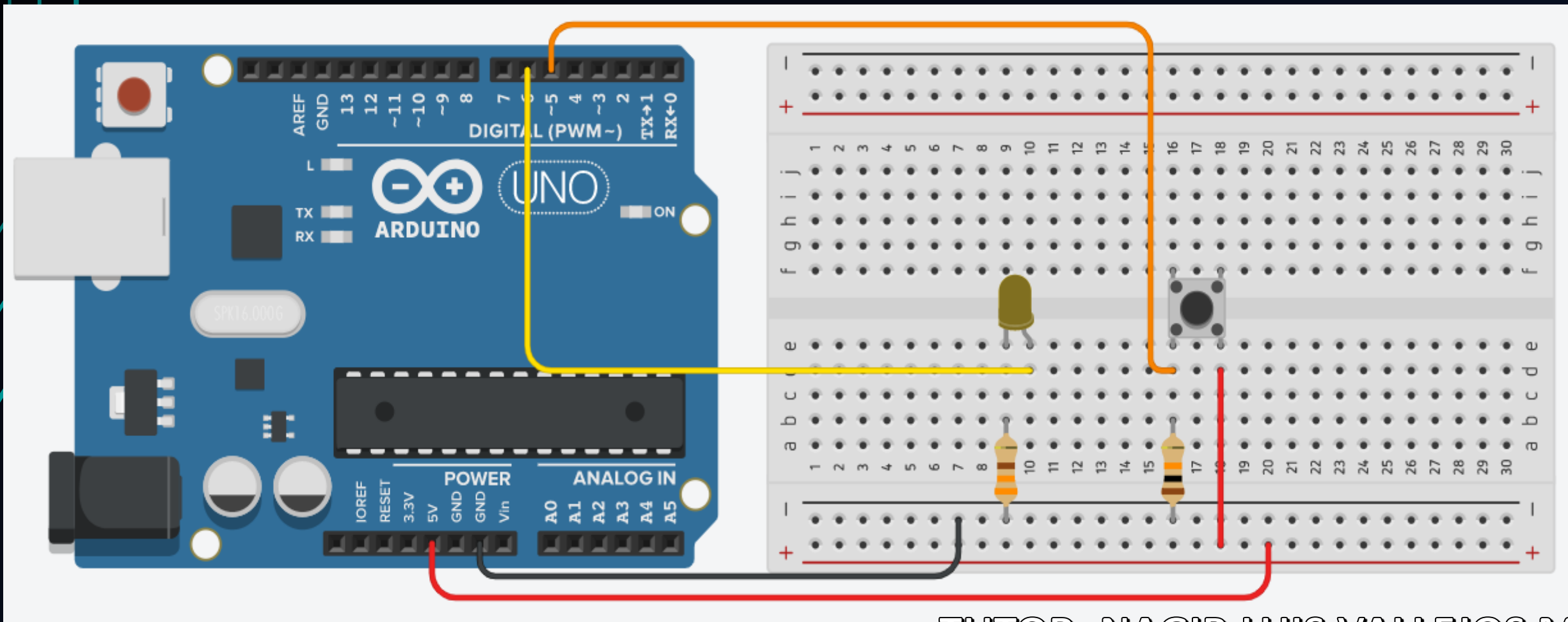


TUTOR: NAGIB LUIS VALLEJOS M.

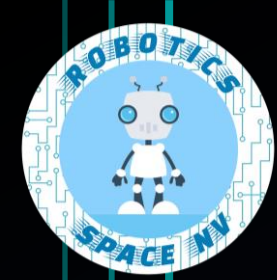


# EJERCICIO 1 – CIRCUITO

Encender un LED a través de un pulsador, empleando una resistencia pull down



TUTOR: NAGIB LUIS VALLEJOS M.



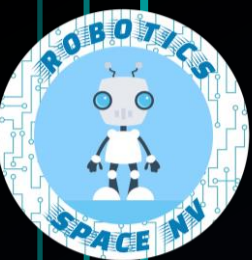
# EJERCICIO 1 – SOLUCIÓN

S3-E1

```
1 int led=6;           //constante
2 int pulsador=5;      //constante
3 int lectura=0;       //variable
4 void setup() {
5     pinMode(pulsador, INPUT);
6     pinMode(led, OUTPUT);
7 }
8
9 void loop() {
10    lectura=digitalRead(pulsador);
11    if(lectura==1) {
12        digitalWrite(led, 1);
13    }
14 }
```

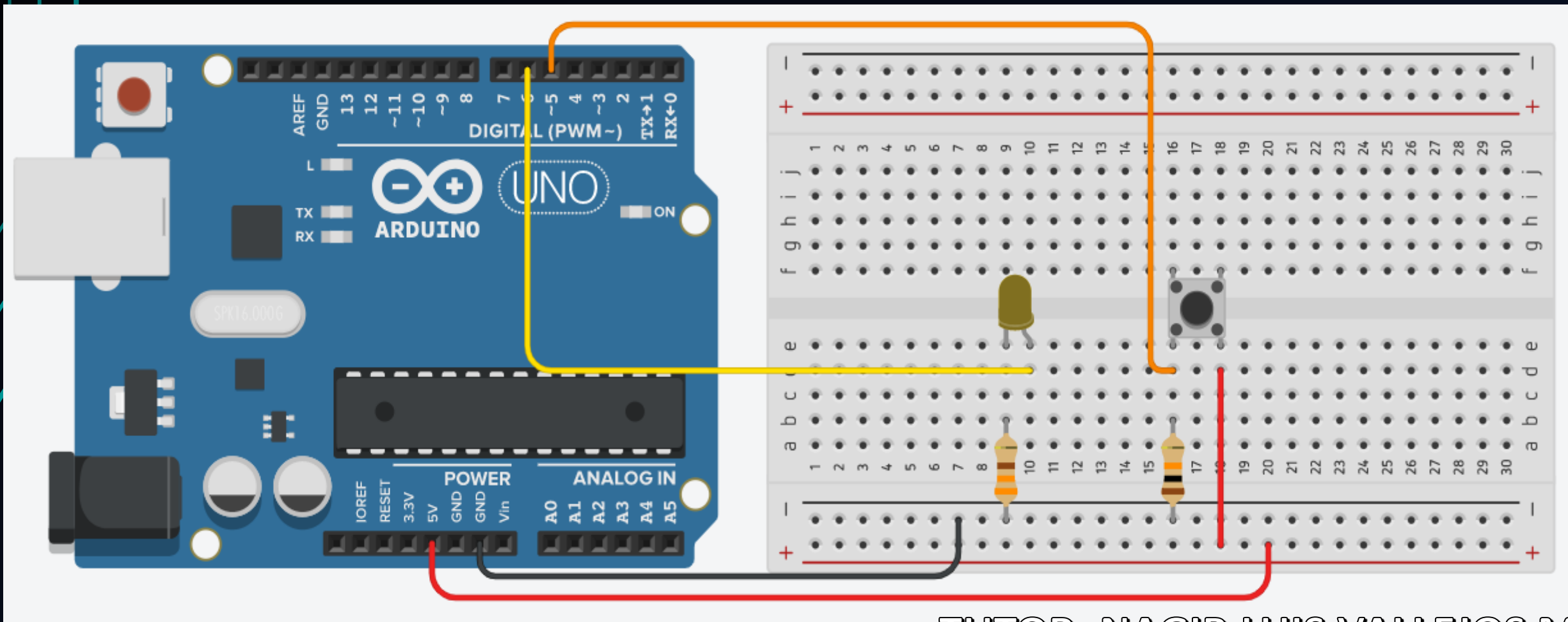
TUTOR: NAGIB LUIS VALLEJOS M.



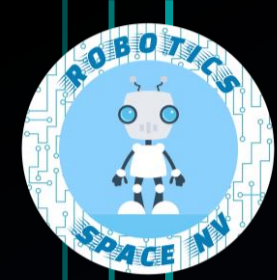


# EJERCICIO 2 – CIRCUITO

Encender y apagar un LED a través de un pulsador, empleando una resistencia pull down



TUTOR: NAGIB LUIS VALLEJOS M.

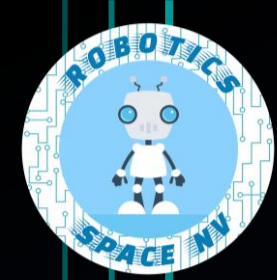


# EJERCICIO 2 – SOLUCIÓN

S3-E2

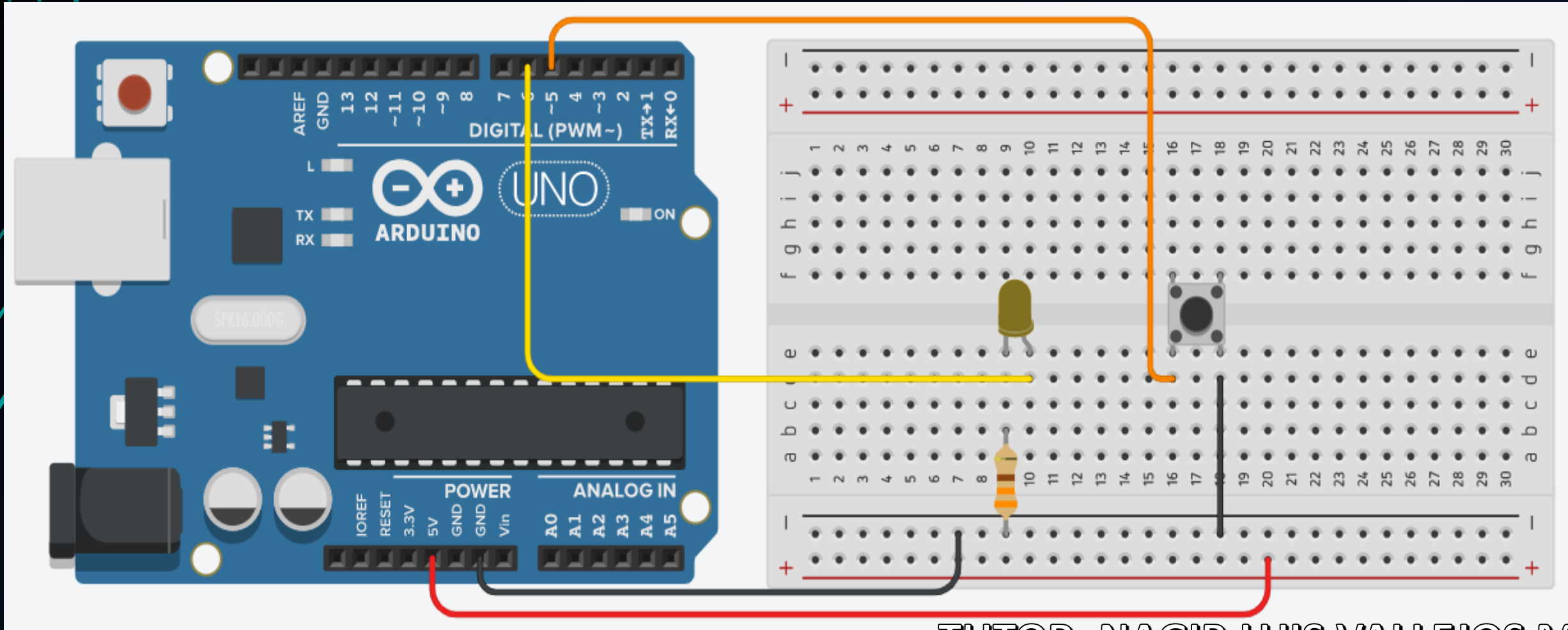
```
1 int led=6;           //constante
2 int pulsador=5;      //constante
3 int lectura=0;       //variable
4 void setup() {
5     pinMode(pulsador, INPUT);
6     pinMode(led, OUTPUT);
7 }
8
9 void loop() {
10    lectura=digitalRead(pulsador);
11    if(lectura==1) {
12        digitalWrite(led, 1);
13    }
14    else{
15        digitalWrite(led, 0);
16    }
17 }
```

TUTOR: NAGIB LUIS VALLEJOS M.

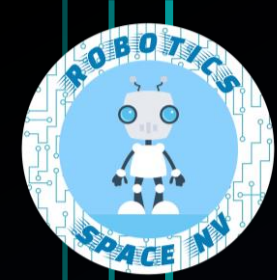


# EJERCICIO 3 – CIRCUITO

Encender y apagar un led a través de un pulsador, empleando una resistencia pull up



TUTOR: NAGIB LUIS VALLEJOS M.

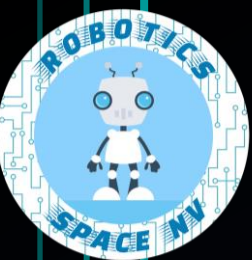


# EJERCICIO 3 – SOLUCIÓN

S3-E3

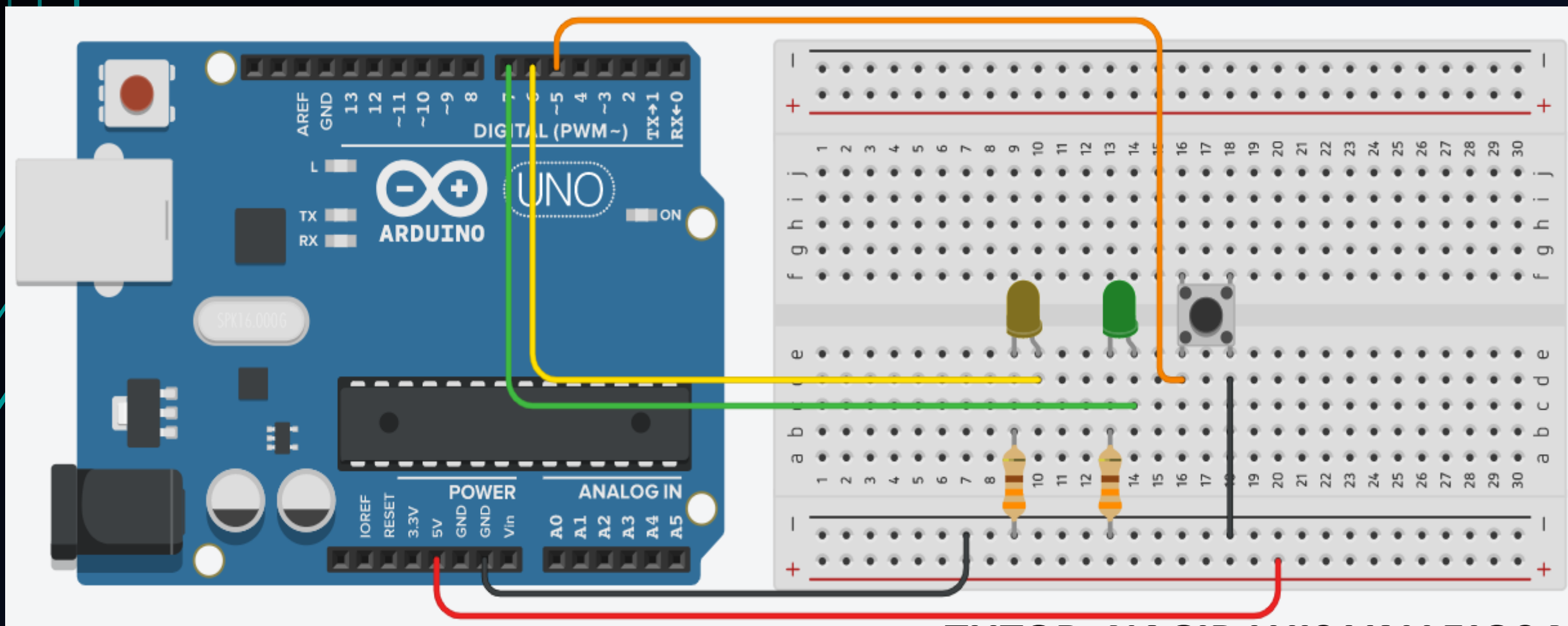
```
1 int led=6;           //constante
2 int pulsador=5;      //constante
3 int lectura=0;       //variable
4 void setup() {
5     pinMode(pulsador, INPUT_PULLUP);
6     pinMode(led, OUTPUT);
7 }
8
9 void loop() {
10    lectura=digitalRead(pulsador);
11    if(lectura==0) {
12        digitalWrite(led,1);
13    }
14    else{
15        digitalWrite(led,0);
16    }
17 }
```

TUTOR: NAGIB LUIS VALLEJOS M.



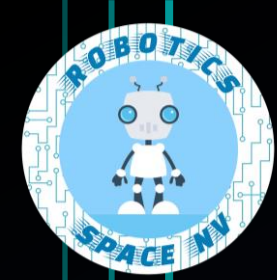
# EJERCICIO 4 – CIRCUITO

Encender y apagar dos leds a través de un pulsador, empleando una resistencia pull up



TUTOR: NAGIB LUIS VALLEJOS M.





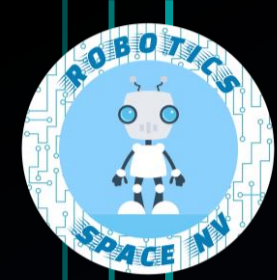
# EJERCICIO 4 – SOLUCIÓN

Encender y apagar dos leds a través de un pulsador, empleando una resistencia pull up

S3-E4

```
1 int ledV=7;
2 int ledA=6;      //constante
3 int pulsador=5; //constante
4 int lectura=0;  //variable
5 void setup() {
6     pinMode(pulsador, INPUT_PULLUP);
7     pinMode(ledA, OUTPUT);
8     pinMode(ledV, OUTPUT);
9 }
10 void loop() {
11     lectura=digitalRead(pulsador);
12     if(lectura==0) {
13         digitalWrite(ledA,1);
14         digitalWrite(ledV,1);
15     }
16     else{
17         digitalWrite(ledA,0);
18         digitalWrite(ledV,0);
19     }
20 }
```

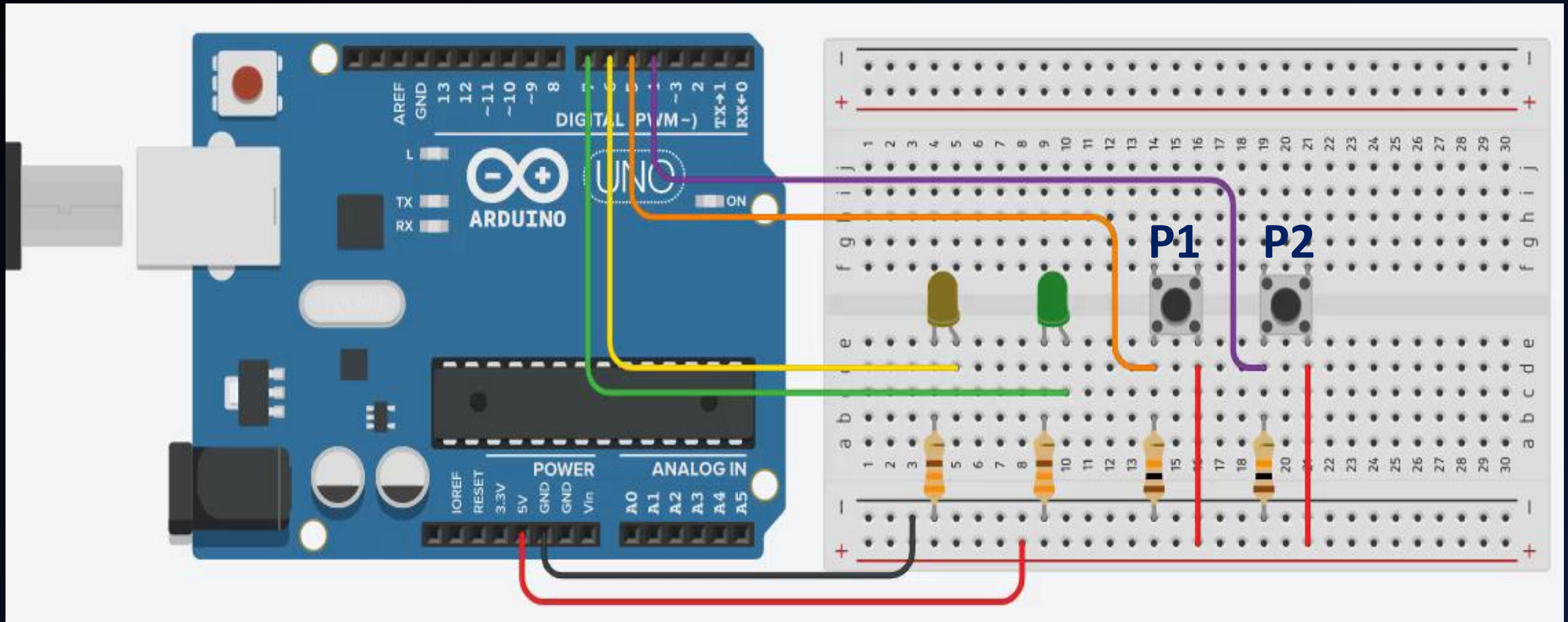
TUTOR: NAGIB LUIS VALLEJOS M.



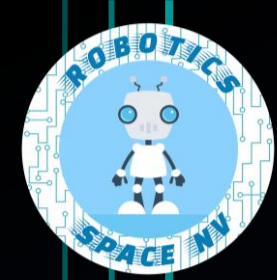
# EJERCICIO 5 – CIRCUITO

Encender y apagar el led amarillo con el pulsador **P1**

Encender y apagar el led verde con el pulsador **P2**



TUTOR: NAGIB LUIS VALLEJOS M.



# EJERCICIO 5 – SOLUCIÓN

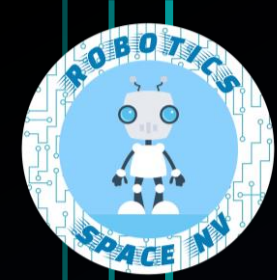
Encender y apagar el led amarillo con el pulsador P1

Encender y apagar el led verde con el pulsador P2

S3-E5

```
1 int ledV=7;
2 int ledA=6;
3 int pulsador=5;
4 int pulsador2=4;
5 int p1=0;
6 int p2=0;
7 void setup() {
8     pinMode(pulsador, INPUT);
9     pinMode(pulsador2, INPUT);
10    pinMode(ledA, OUTPUT);
11    pinMode(ledV, OUTPUT);
12 }
13 void loop() {
14     p1=digitalRead(pulsador);
15     if (p1==1) {
16         digitalWrite(ledA, 1);
17     }
18     else{
19         digitalWrite(ledA, 0);
20     }
21     p2=digitalRead(pulsador2);
22     if (p2==1) {
23         digitalWrite(ledV, 1);
24     }
25     else{
26         digitalWrite(ledV, 0);
27     }
28 }
```

TUTOR: NAGIB LUIS VALLEJOS M.

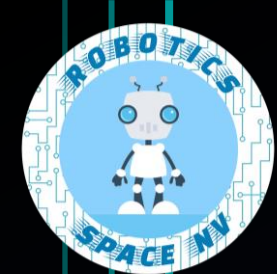


# ALGEBRA DE BOOLE

Cuando trabajamos con electrónica digital una de las áreas más importantes es el *Algebra de Boole*, ya que este nos permite emplear el uso de conectivas lógicas como ser:

PREMISAS		CONECTIVAS LÓGICAS				
p	q	$\wedge$	$\vee$	$\rightarrow$	$\leftrightarrow$	<u><math>\neg</math></u>
0	0	0	0	1	1	0
0	1	0	1	1	0	1
1	0	0	1	0	0	1
1	1	1	1	1	1	0

TUTOR: NAGIB LUIS VALLEJOS M.



# CONDICIONAL COMPUESTA

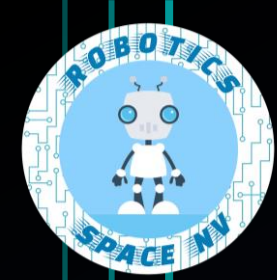
Las estructuras condicionales compuestas son aquellas que cuentan con dos o más condiciones. A través de ello, para ejecutar un código determinado, si o si debe cumplirse 1 o más condiciones. Por ejemplo:

Dado un  $X$ , verificar si  $X$  es par y también es primo. Tomaremos como  $X=2$  y  $X=4$

Si  $X=2$  entonces, es par y también es primo. **V**

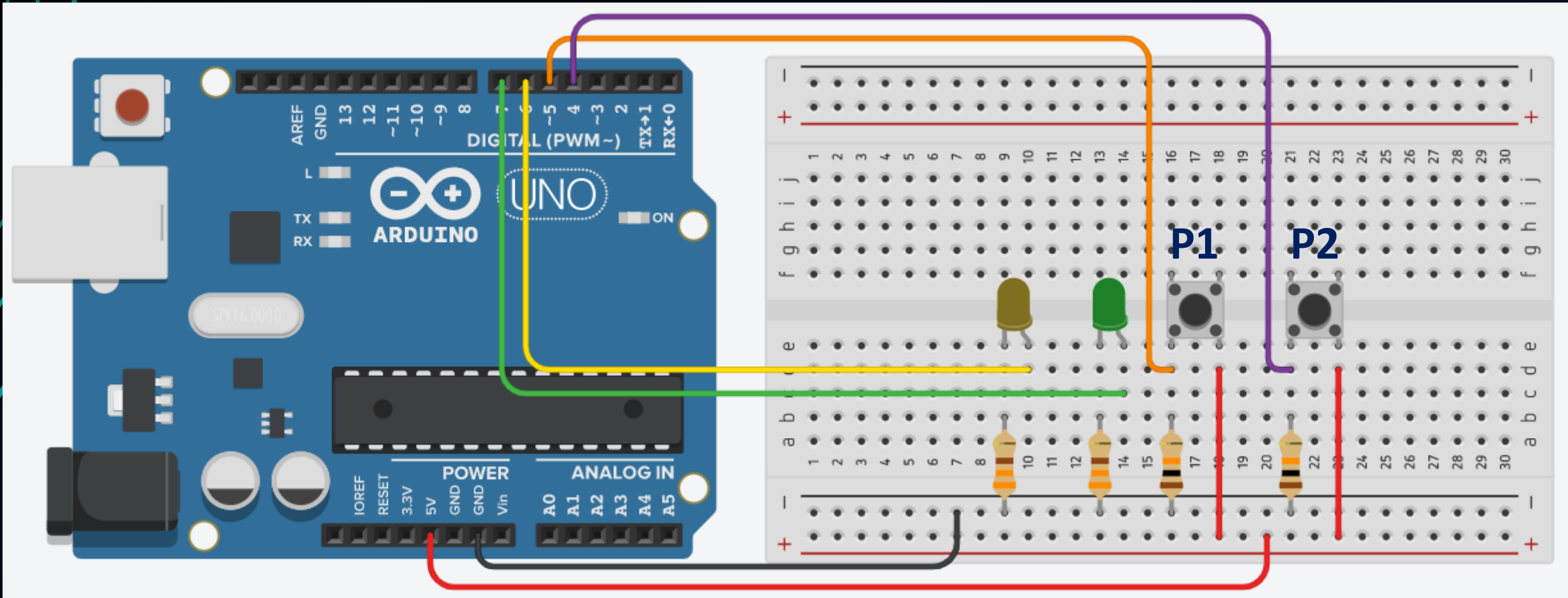
Si  $X=4$  entonces, es par pero no es primo. **F**



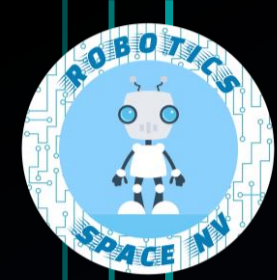


# EJEMPLO 6 – CIRCUITO

Prender 2 leds al mismo tiempo, si y solo si P1 y P2 están presionados, de lo contrario, ambos leds deben permanecer apagados



TUTOR: NAGIB LUIS VALLEJOS M.



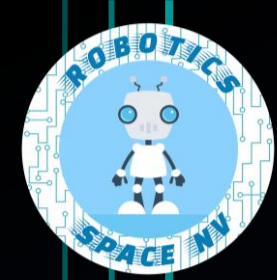
# EJEMPLO 6 – SOLUCIÓN

Prender 2 leds al mismo tiempo, si y solo si P1 y P2 están presionados, de lo contrario, ambos leds deben permanecer apagados

S3-E6

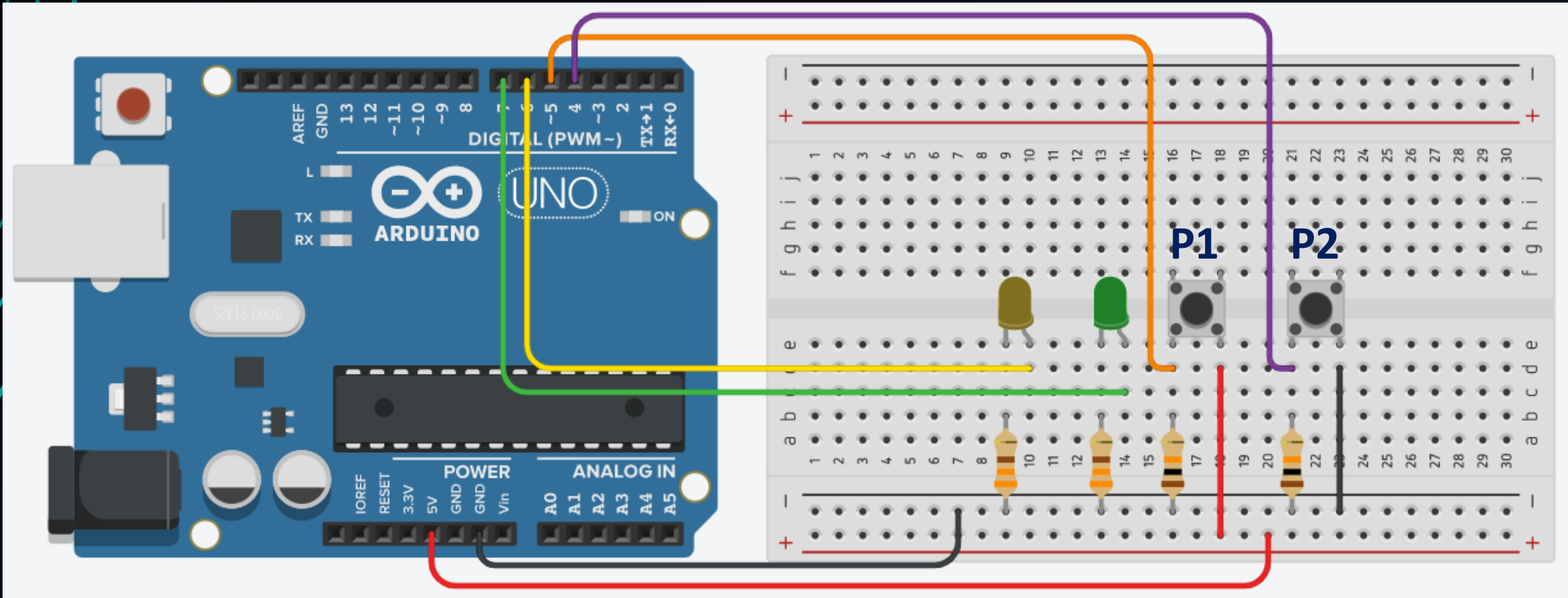
```
1 int ledV=7;
2 int ledA=6;
3 int pulsador=5;
4 int pulsador2=4;
5 int p1=0;
6 int p2=0;
7 void setup() {
8     pinMode(pulsador, INPUT);
9     pinMode(pulsador2, INPUT);
10    pinMode(ledA, OUTPUT);
11    pinMode(ledV, OUTPUT);
12 }
13 void loop() {
14     p1=digitalRead(pulsador);
15     p2=digitalRead(pulsador2);
16     if(p1==1 and p2==1){
17         digitalWrite(ledA,1);
18         digitalWrite(ledV,1);
19     }
20     else{
21         digitalWrite(ledA,0);
22         digitalWrite(ledV,0);
23     }
24 }
```

TUTOR: NAGIB LUIS VALLEJOS M.

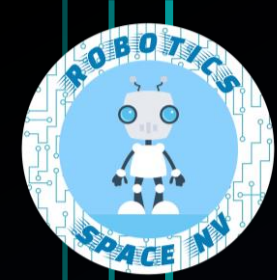


# EJEMPLO 7 – CIRCUITO

Prender 2 LED's si P1 o P2 están presionados, de lo contrario, ambos leds deben permanecer apagados



TUTOR: NAGIB LUIS VALLEJOS M.



# EJEMPLO 7 – SOLUCIÓN

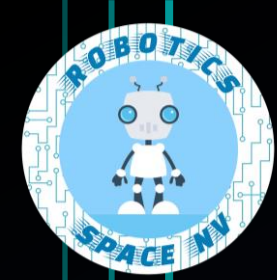
Prender 2 LED's si P1 o P2 están presionados, de lo contrario, ambos leds deben permanecer apagados

S3-E7

```
1 int ledV=7;
2 int ledA=6;
3 int pulsador=5;
4 int pulsador2=4;
5 int p1=0;
6 int p2=0;
7 void setup() {
8     pinMode(pulsador, INPUT);
9     pinMode(pulsador2, INPUT);
10    pinMode(ledA, OUTPUT);
11    pinMode(ledV, OUTPUT);
12 }
13 void loop() {
14     p1=digitalRead(pulsador);
15     p2=digitalRead(pulsador2);
16     if(p1==1 or p2==1){
17         digitalWrite(ledA, 1);
18         digitalWrite(ledV, 1);
19     }
20     else{
21         digitalWrite(ledA, 0);
22         digitalWrite(ledV, 0);
23     }
24 }
```

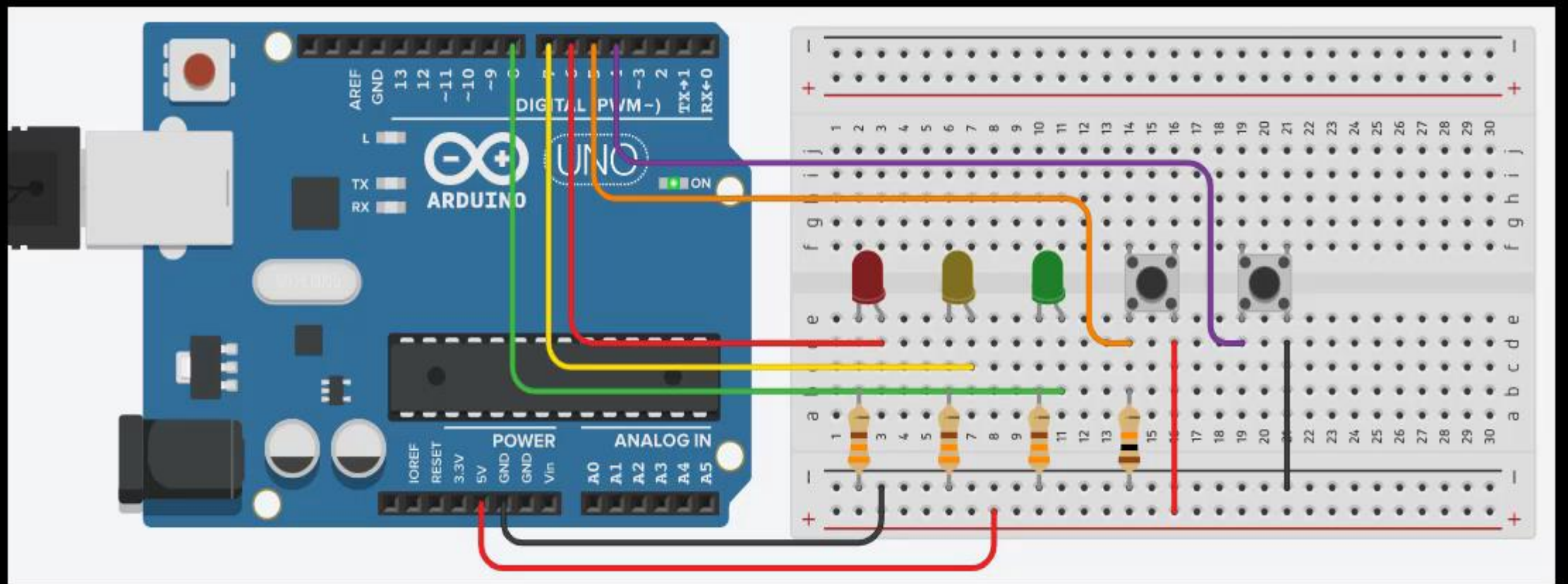
TUTOR: NAGIB LUIS VALLEJOS M.





# RETO CLASE

Encender y apagar 3 leds empleando todo el avance de la clase, usaremos resistencias pull down, pull up, 2 pulsadores y emplear condicionales simples y dobles si es necesario. El video de muestra se encuentra en el repositorio



TUTOR: NAGIB LUIS VALLEJOS M.



# CONTACTOS



SUSCRÍBETE



(+591) 63096640



robotics.space.nv@gmail.com



fb.me/RoboticsSpaceNV



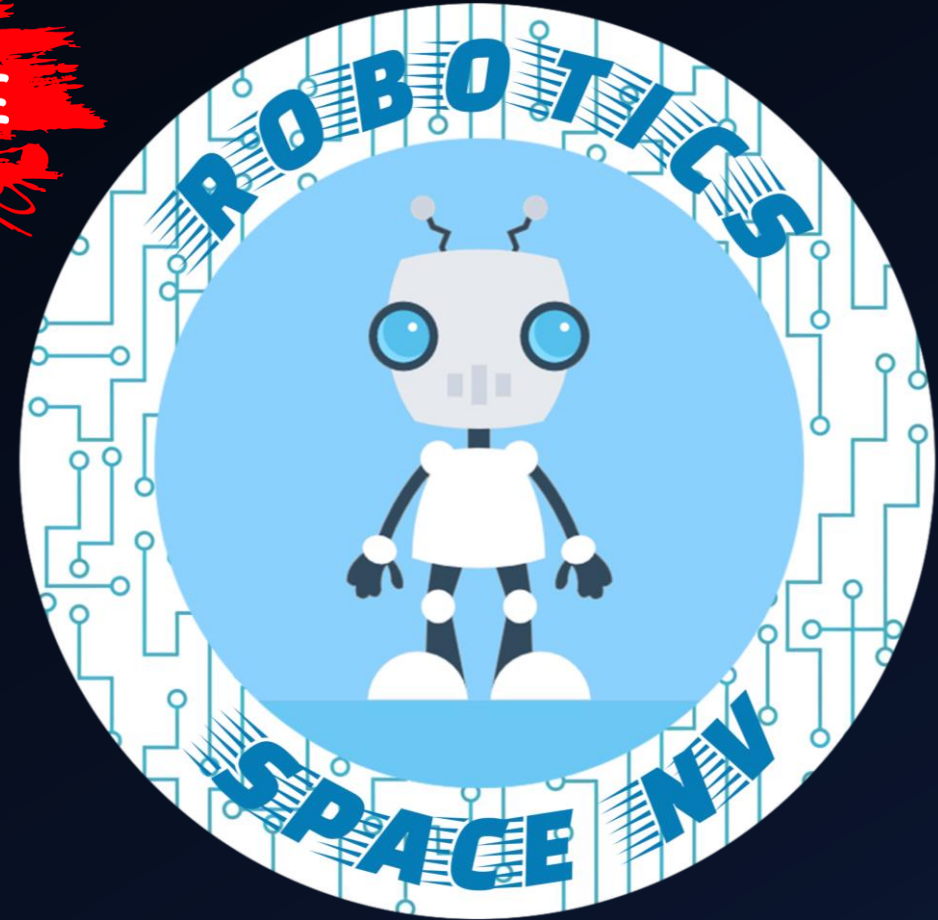
@NagibVallejos



Robotics Space NV



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



TUTOR: NAGIB LUIS VALLEJOS M.