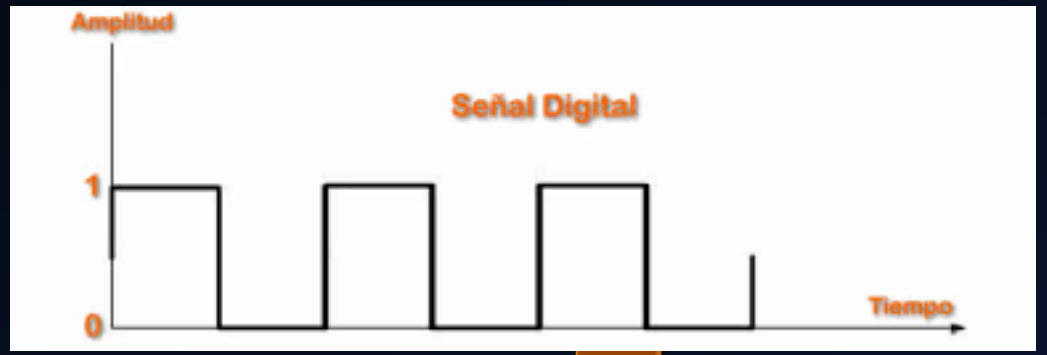
Suscríbete

Clase 4

ENTRADAS DIGITALES



TUTOR:NAGIB LUIS VALLEJOS M.

ENTRADAS DIGITALES

Una **entrada digital** es aquella que permite enviar una señal del ambiente hacia arduino. Los valores cambian entre 0-1.

Los pines digitales de arduino vienen configurados como entrada por defecto, por lo cual no es necesario declararlos, sin embargo es recomendable para entender su funcionamiento.

Para declararla se utiliza la siguiente sintaxis:

pinMode(pin,INPUT);

TUTOR:NAGIB LUIS VALLEJOS M.

ENTRADAS DIGITALES

¿Cuánta corriente consume una entrada digital?

La corriente empleada es mínima y equivalente a 100 MΩ. Esto la hace útil para poder recabar el estado de un led o leer información a través de un sensor.

Si el consumo es mínimo, ¿por qué se realiza el uso de una resistencia?

La resistencia permite eliminar el ruido eléctrico del entorno por lo cual sin el uso de esta, el pin digital configurado como entrada llega a tomar valores aleatorios entre 0 y 1

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

TUTOR:NAGIB LUIS VALLEJOS M.

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.

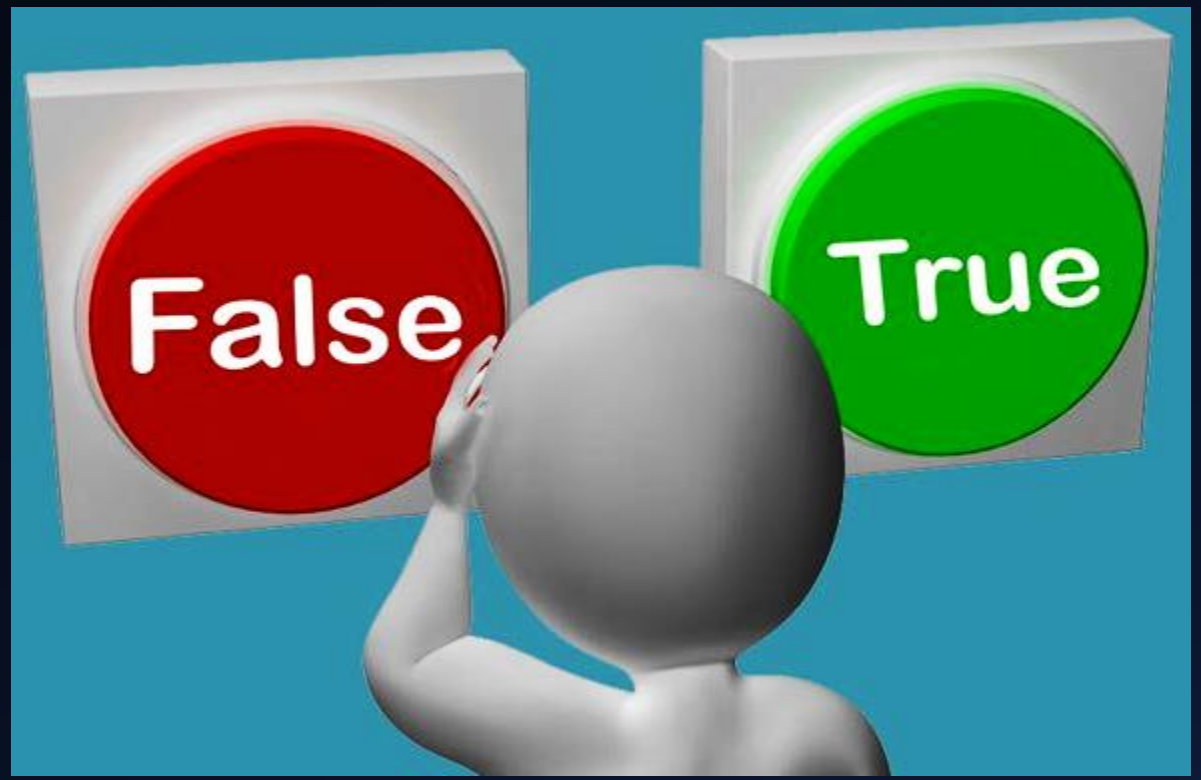
ESTRUCTURAS DE CONTROL

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

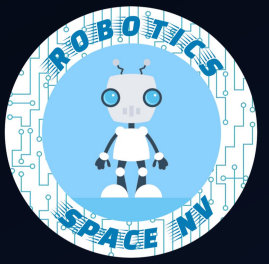
Permite indicar la veracidad de una condición, por lo cual emplea operadores comparativos los cuales son:

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



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*

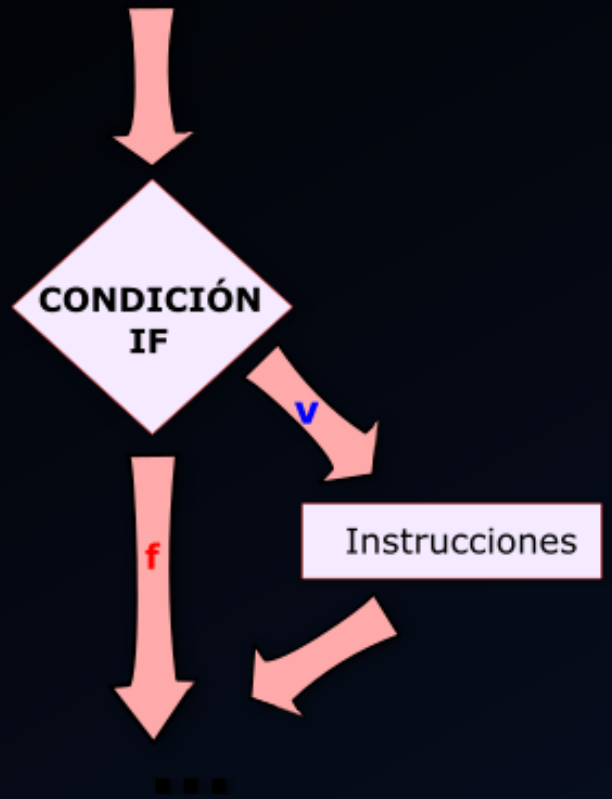
Donde:

V = 1 o HIGH.

F = 0 o LOW.

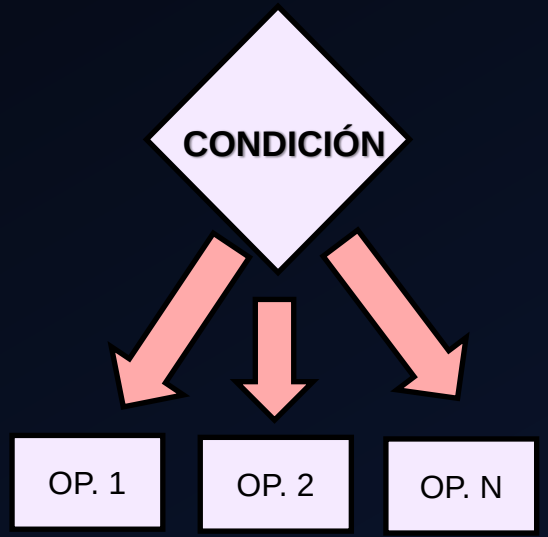
TUTOR:NAGIB LUIS VALLEJOS M.

TIPOS DE CONDICIONALES



CONDICIONAL SIMPLE

CONDICIONAL DOBLE



CONDICIONAL MÚLTIPLE

TUTOR:NAGIB LUIS VALLEJOS M.

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 KΩ.

TIPOS DE PULSADORES

TUTOR:NAGIB LUIS VALLEJOS M.

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

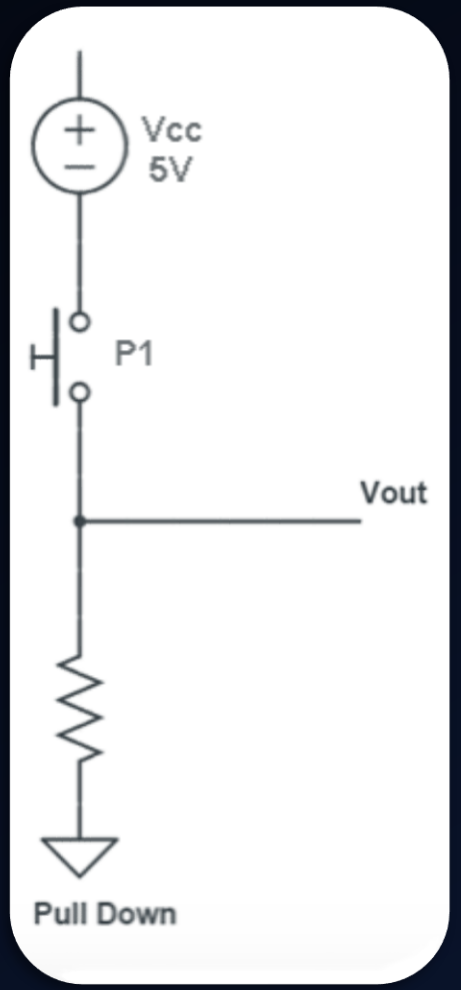
En arduino el uso de variables y constantes permiten dar un mayor entendimiento a nuestros programas ya que nos permiten identificar los componentes de uso y pines donde se encuentran conectados, sin necesidad de tener que ver el circuito de manera física.

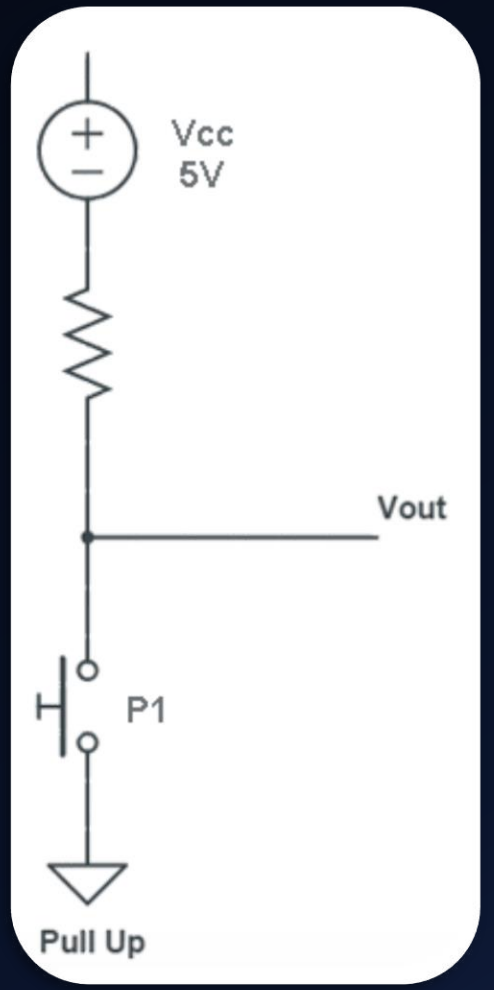
TUTOR:NAGIB LUIS VALLEJOS M.

FUNCIÓN INPUT E INPUT_PULLUP

La función **INPUT** permite habilitar un pin como entrada y requiere del uso de usa 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***

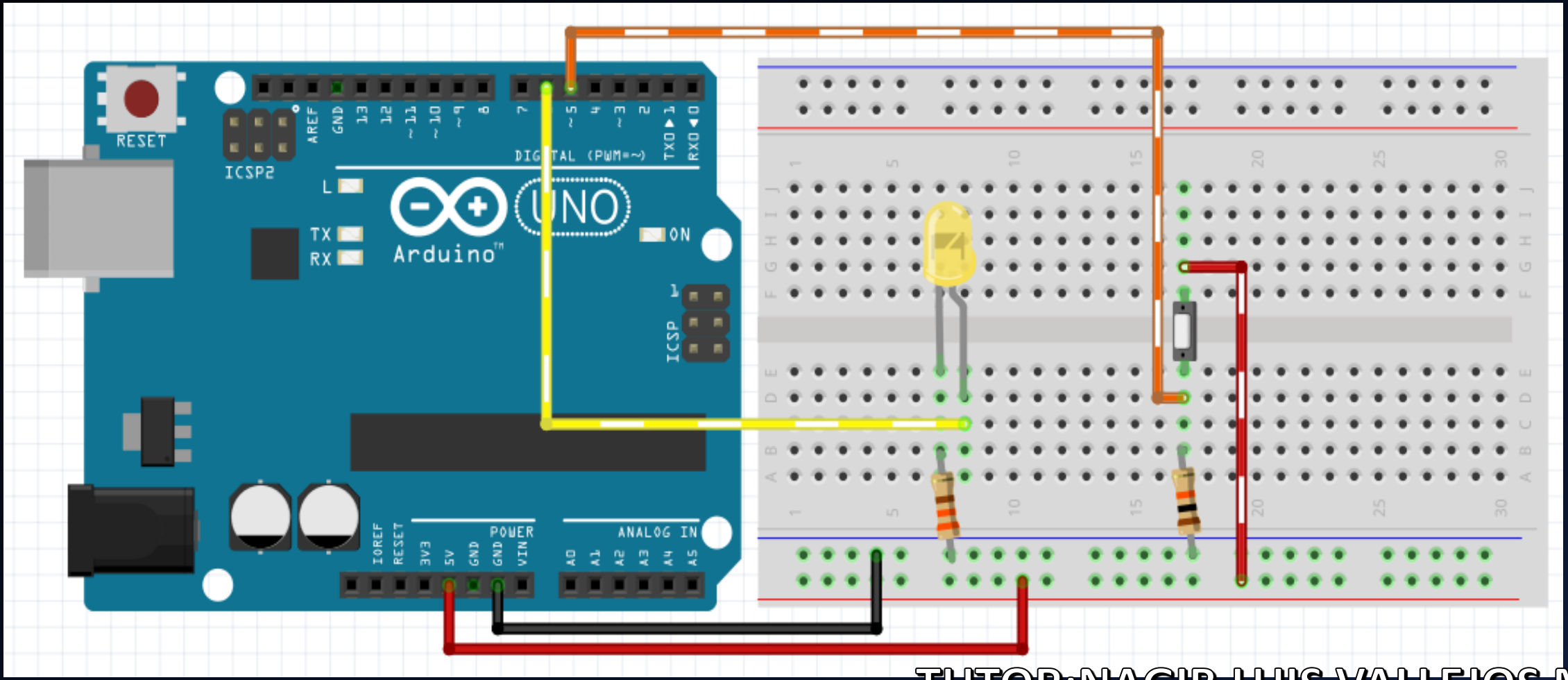




TUTOR:NAGIB LUIS VALLEJOS M.

EJERCICIO 1 – CIRCUITO

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



TUTOR:NAGIB LUIS VALLEJOS M.

EJERCICIO 1 – SOLUCIÓN

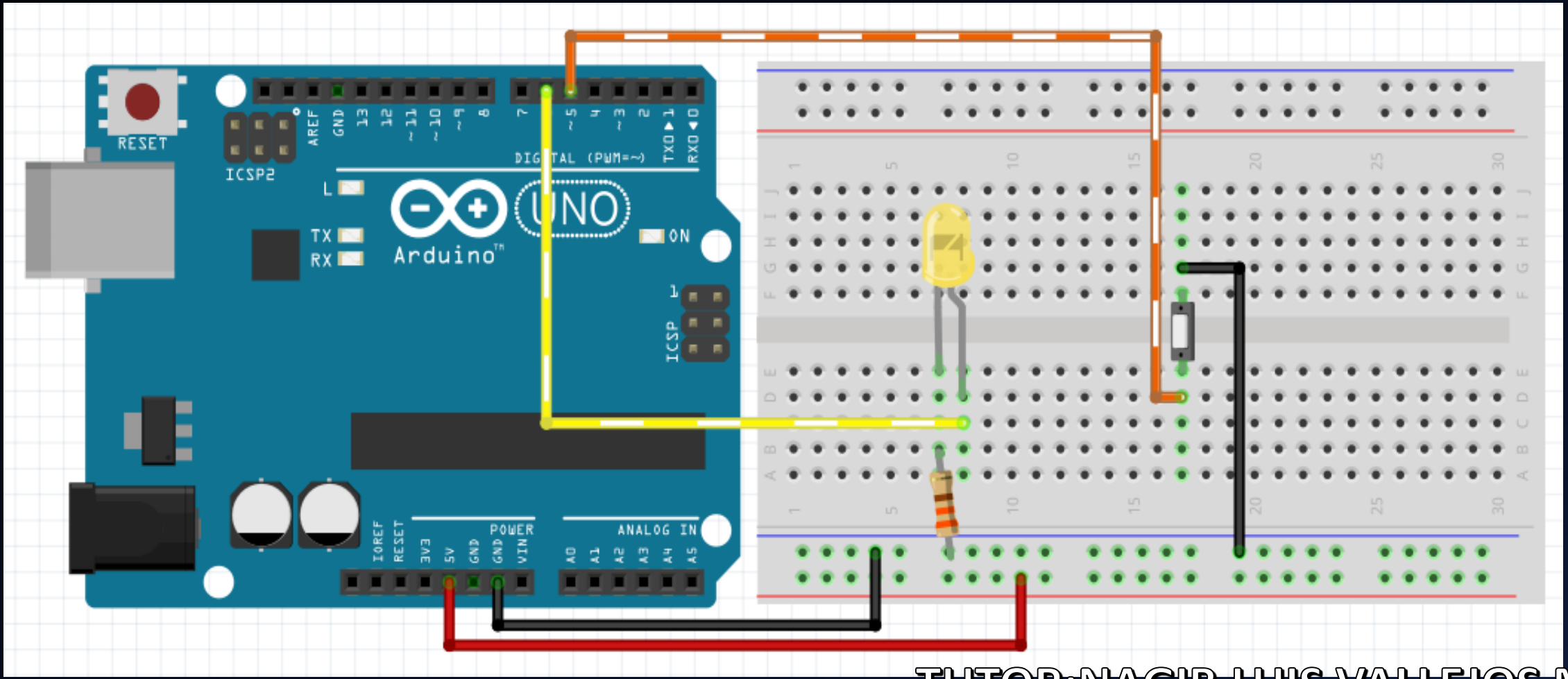
S4-E1

```
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   else{
15     digitalWrite(led,0);
16   }
17 }
```

TUTOR:NAGIB LUIS VALLEJOS M.

EJERCICIO 2 – CIRCUITO

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



TUTOR:NAGIB LUIS VALLEJOS M.

EJERCICIO 2 – SOLUCIÓN

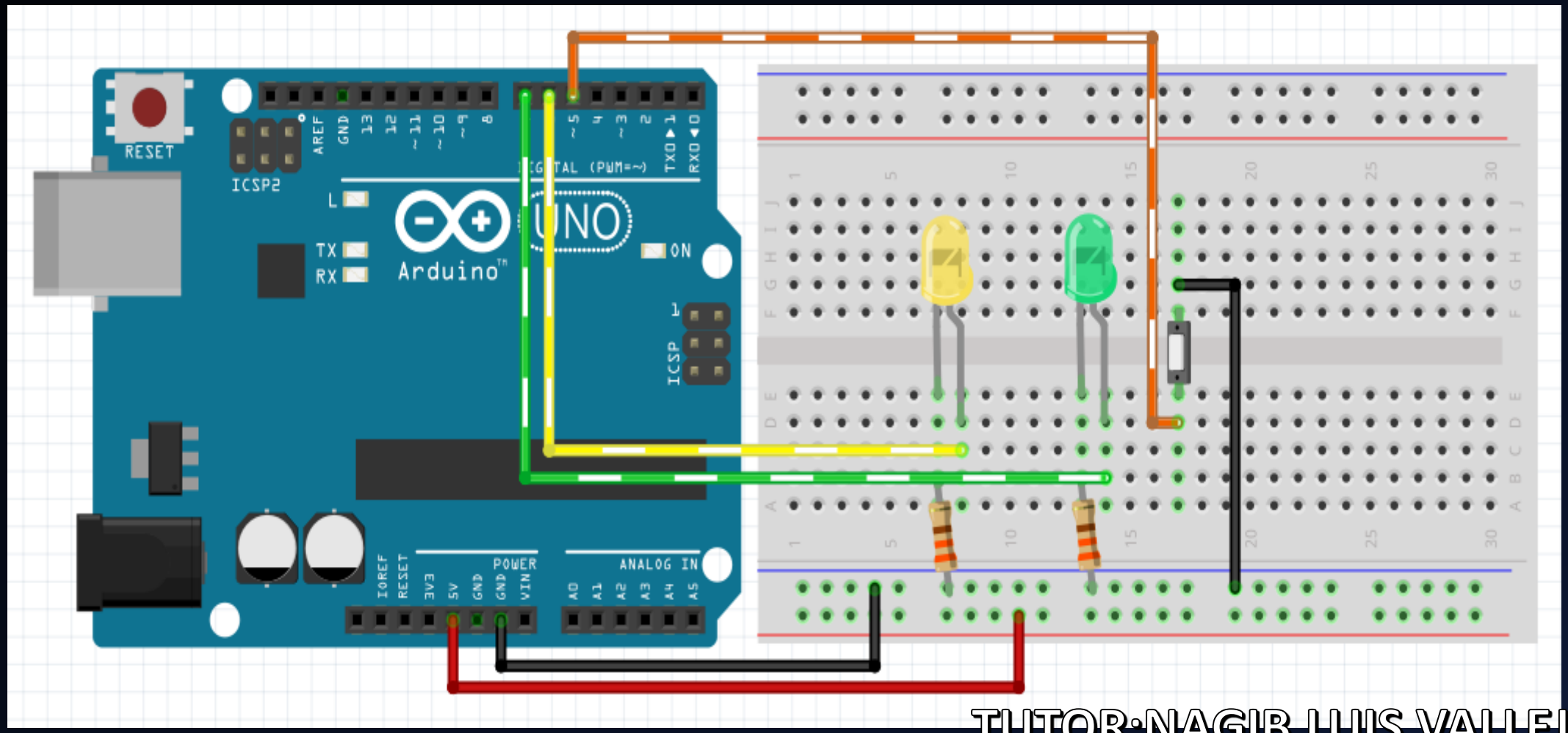
S4-E2

```
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  void loop(){
9    lectura=digitalRead(pulsador);
10   if(lectura==0){
11     digitalWrite(led,1);
12   }
13   else{
14     digitalWrite(led,0);
15   }
16 }
```

TUTOR:NAGIB LUIS VALLEJOS M.

EJERCICIO 3 – CIRCUITO

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



TUTOR:NAGIB LUIS VALLEJOS M.

EJERCICIO 3 – SOLUCIÓN

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

S4-E3

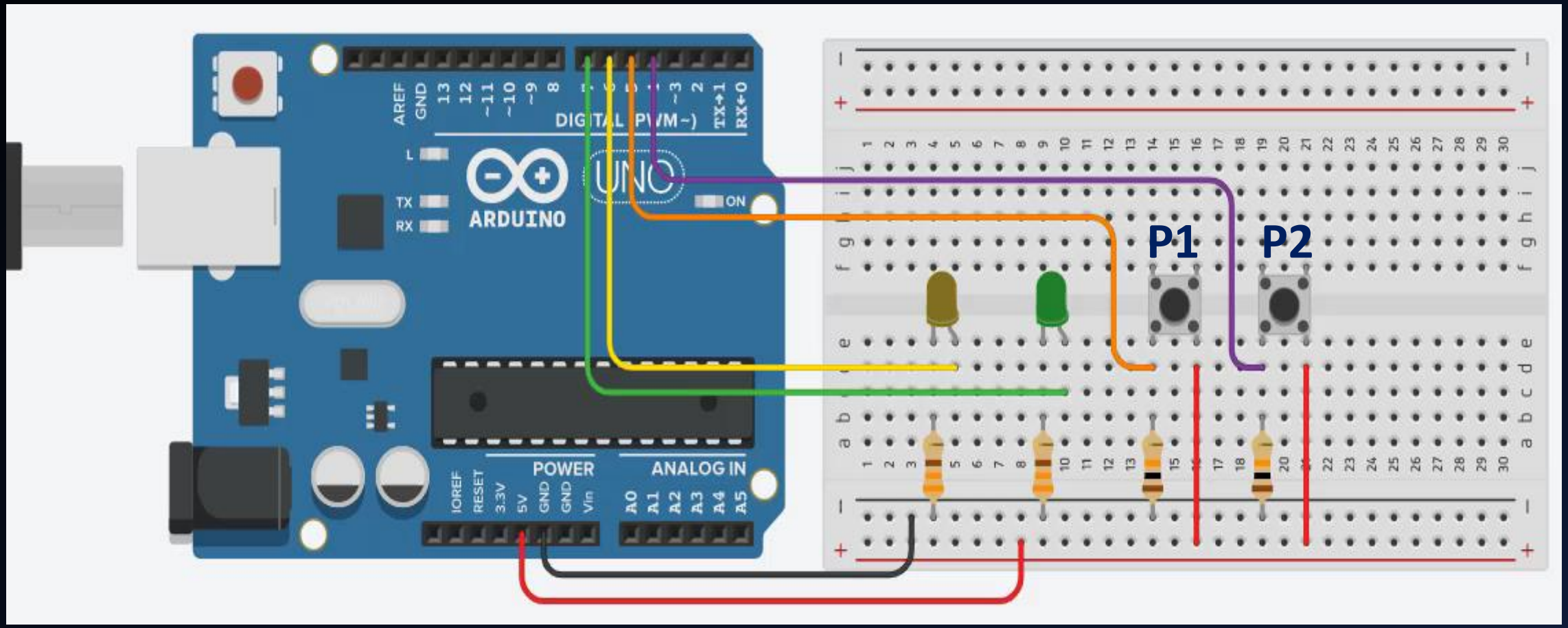
```
int ledV=7;
int ledA=6;       //constante
int pulsador=5; //constante
int lectura=0;  //variable
void setup(){
  pinMode(pulsador,INPUT_PULLUP);
  pinMode(ledA,OUTPUT);
  pinMode(ledV,OUTPUT);
}
void loop(){
  lectura=digitalRead(pulsador);
  if(lectura==0){
    digitalWrite(ledA,1);
    digitalWrite(ledV,1);
  }
  else{
    digitalWrite(ledA,0);
    digitalWrite(ledV,0);
  }
}
```

TUTOR:NAGIB LUIS VALLEJOS M.

EJERCICIO 4 – 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 4 – SOLUCIÓN

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

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

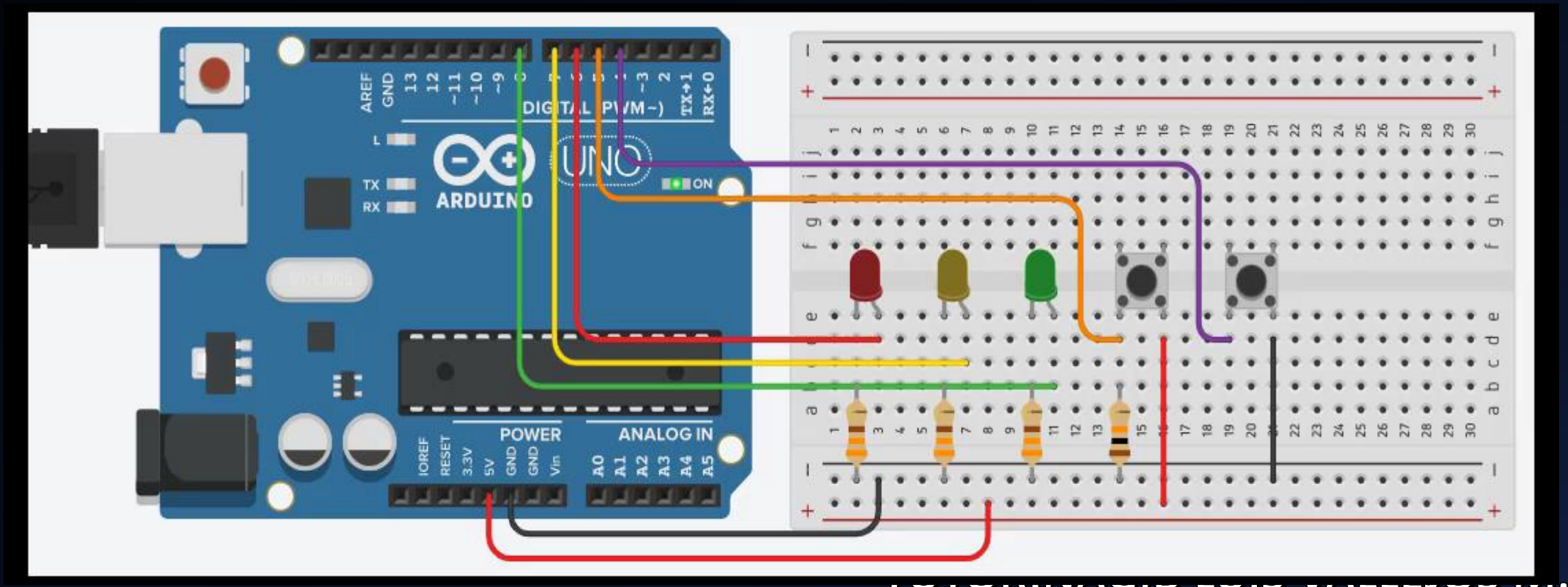
S4-E4

```
int ledV=7;
int ledA=6;
int pulsador=5;
int pulsador2=4;
int p1=0;
int p2=0;
void setup(){
  pinMode(pulsador,INPUT);
  pinMode(pulsador2,INPUT);
  pinMode(ledA,OUTPUT);
  pinMode(ledV,OUTPUT);
}
void loop(){
  p1=digitalRead(pulsador);
  if(p1==1){
    digitalWrite(ledA,1);
  }
  else{
    digitalWrite(ledA,0);
  }
  p2=digitalRead(pulsador2);
  if(p2==1){
    digitalWrite(ledV,1);
  }
  else{
    digitalWrite(ledV,0);
  }
}
```

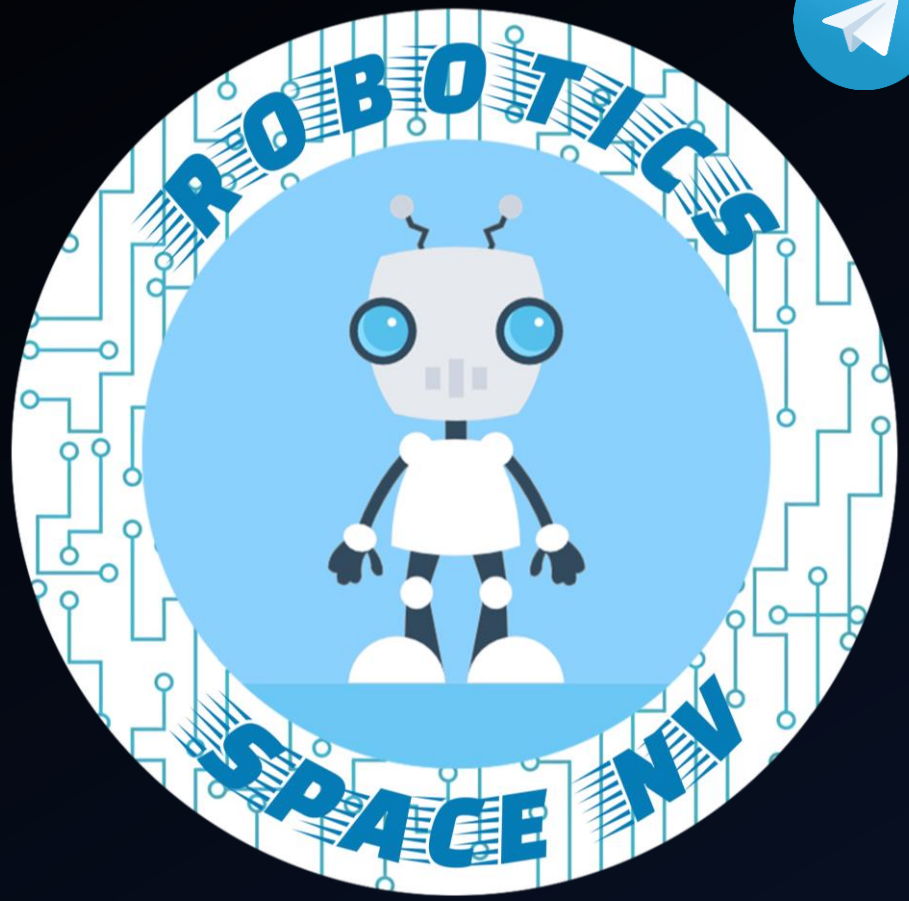
TUTOR:NAGIB LUIS VALLEJOS M.

EJERCICIO PRÁCTICO

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



CONTACTOS



(+591) 63096640

robotics.space.nv@gmail.com

fb.me/RoboticsSpaceNV

@NagibVallejos

Robotics Space NV

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

TUTOR:NAGIB LUIS VALLEJOS M.