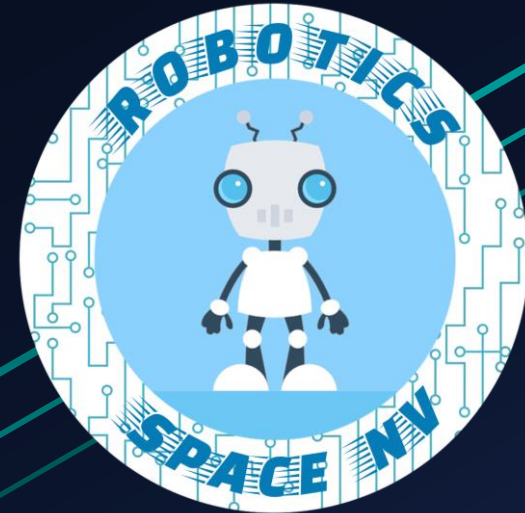
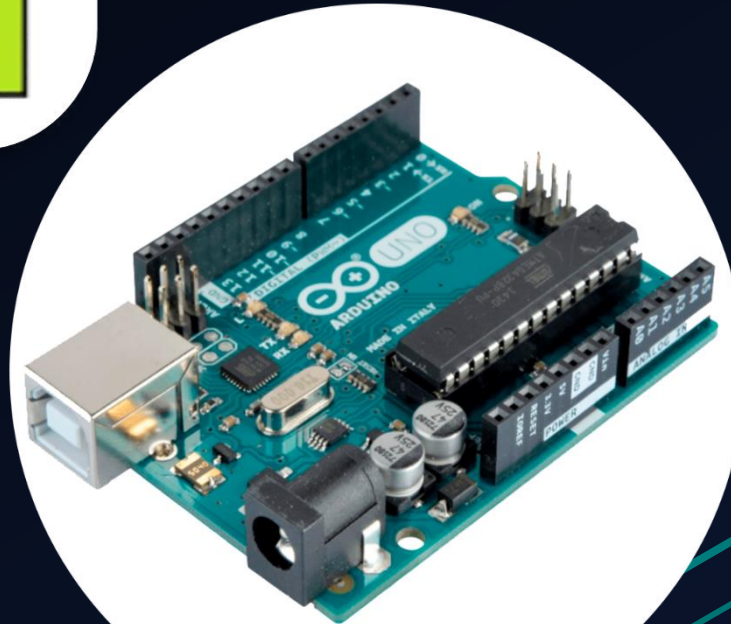
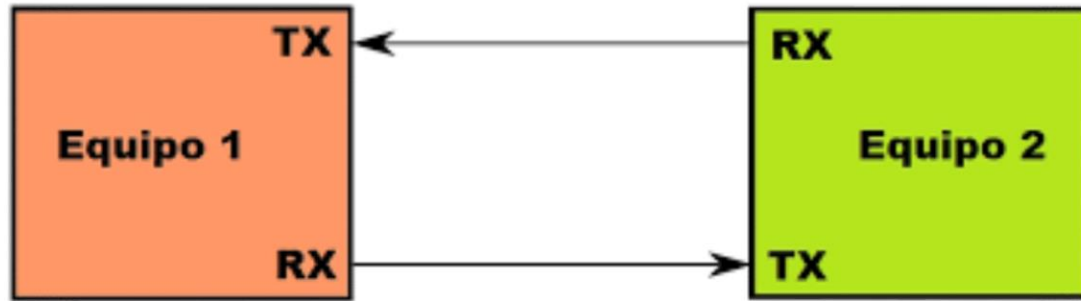


# Clase 23

# PROTOCOLO UART

Suscríbete



TUTOR: NAGIB LUIS VALLEJOS M.

# PROTOCOLO UART

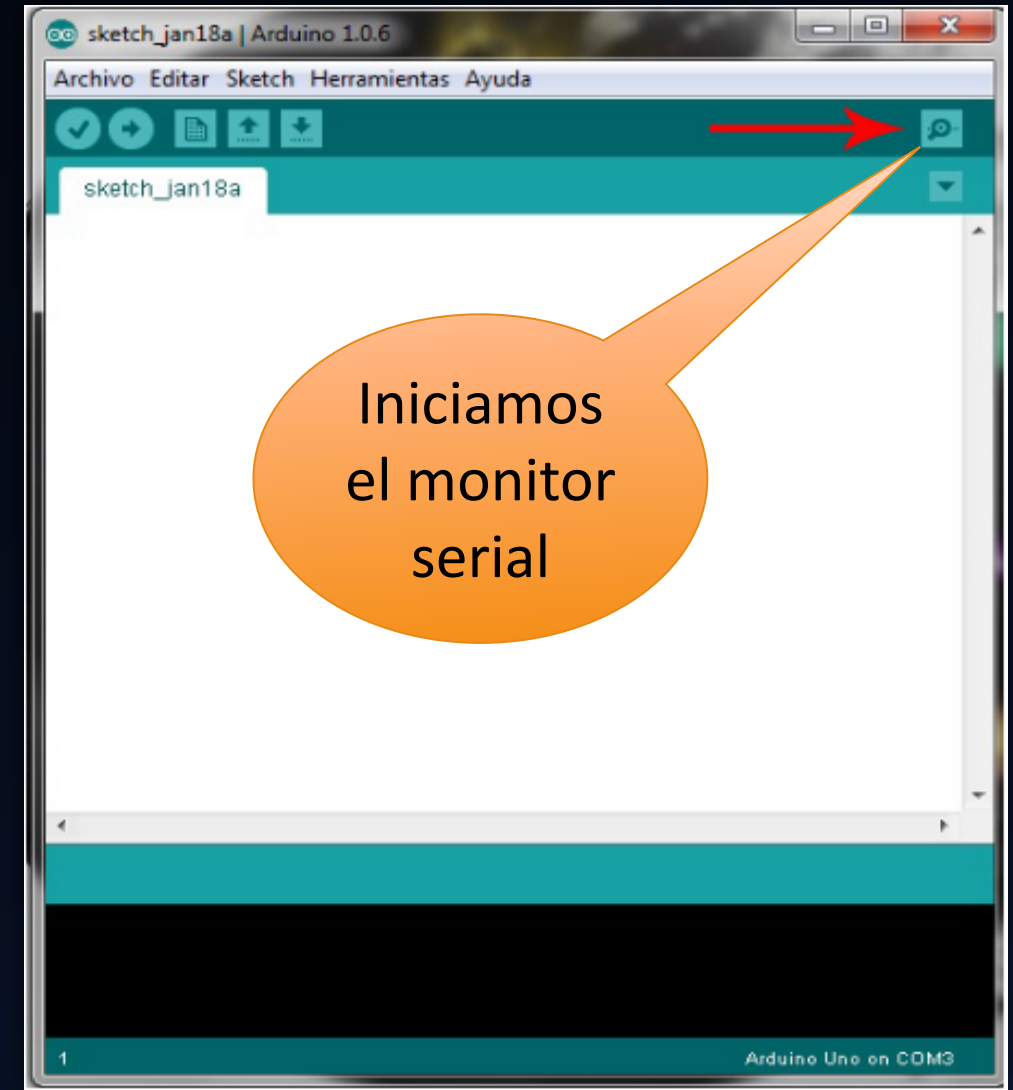
La **comunicación serial** es una “interfaz de comunicación de datos”.

Nos permite visualizar datos y también realizar transferencia de información con diferentes dispositivos”.

Tiene dos funcionalidades:

TX      ►      Transmisión de datos.

RX      ◄      Recepción de datos.



# UART – SALIDA DE DATOS

## TX ► TRANSMISIÓN DE DATOS

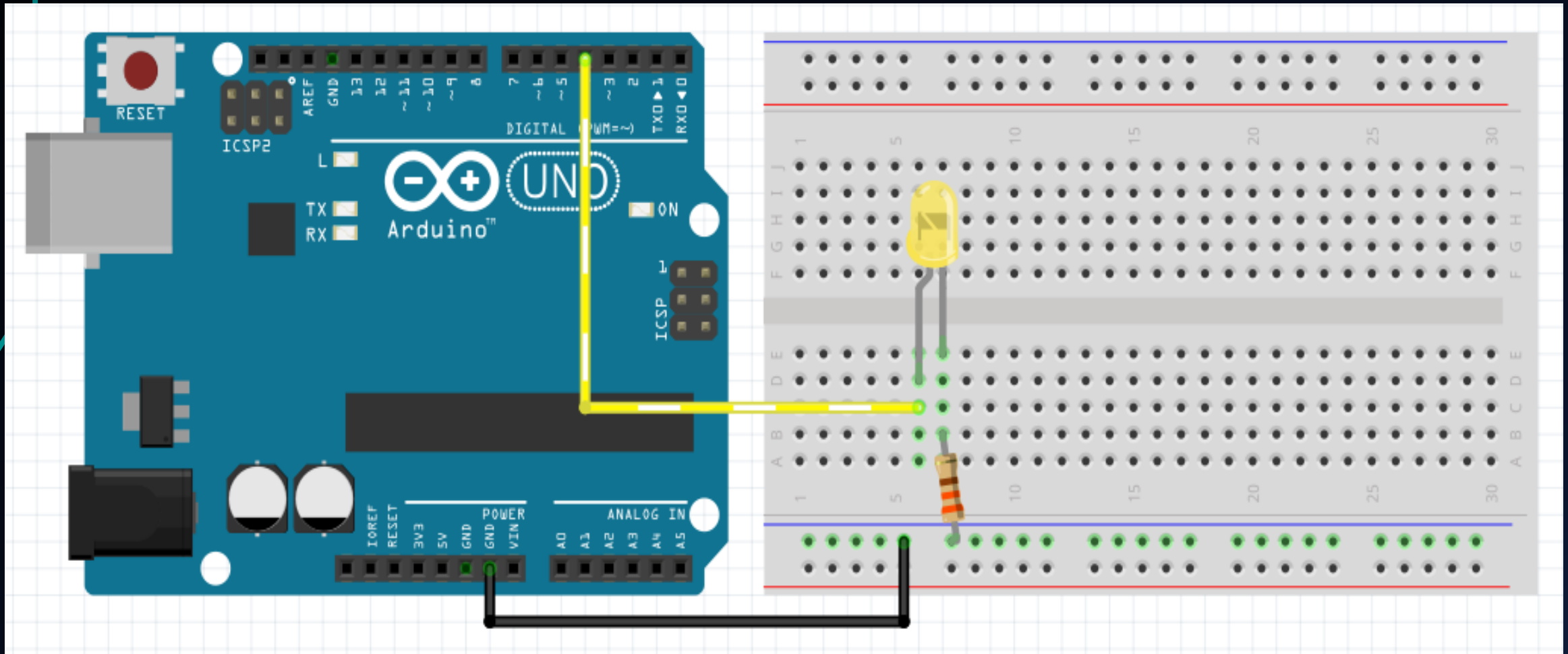
Para poder realizar la transmisión de datos, basta con inicializar la librería **Serial** con una velocidad de 9600 baudios. Esto debido a que dicha velocidad es la tasa de transferencia manejada por defecto.

Su sintaxis es la siguiente:

```
Serial.begin(9600);
```

# EJERCICIO 1 – CIRCUITO

Realizar el encendido y apagado de un led a razón de 1 segundo. Cuando el led se encienda imprimir: **“Encendido”** e imprimir **“Apagado”** cuando el led se apague



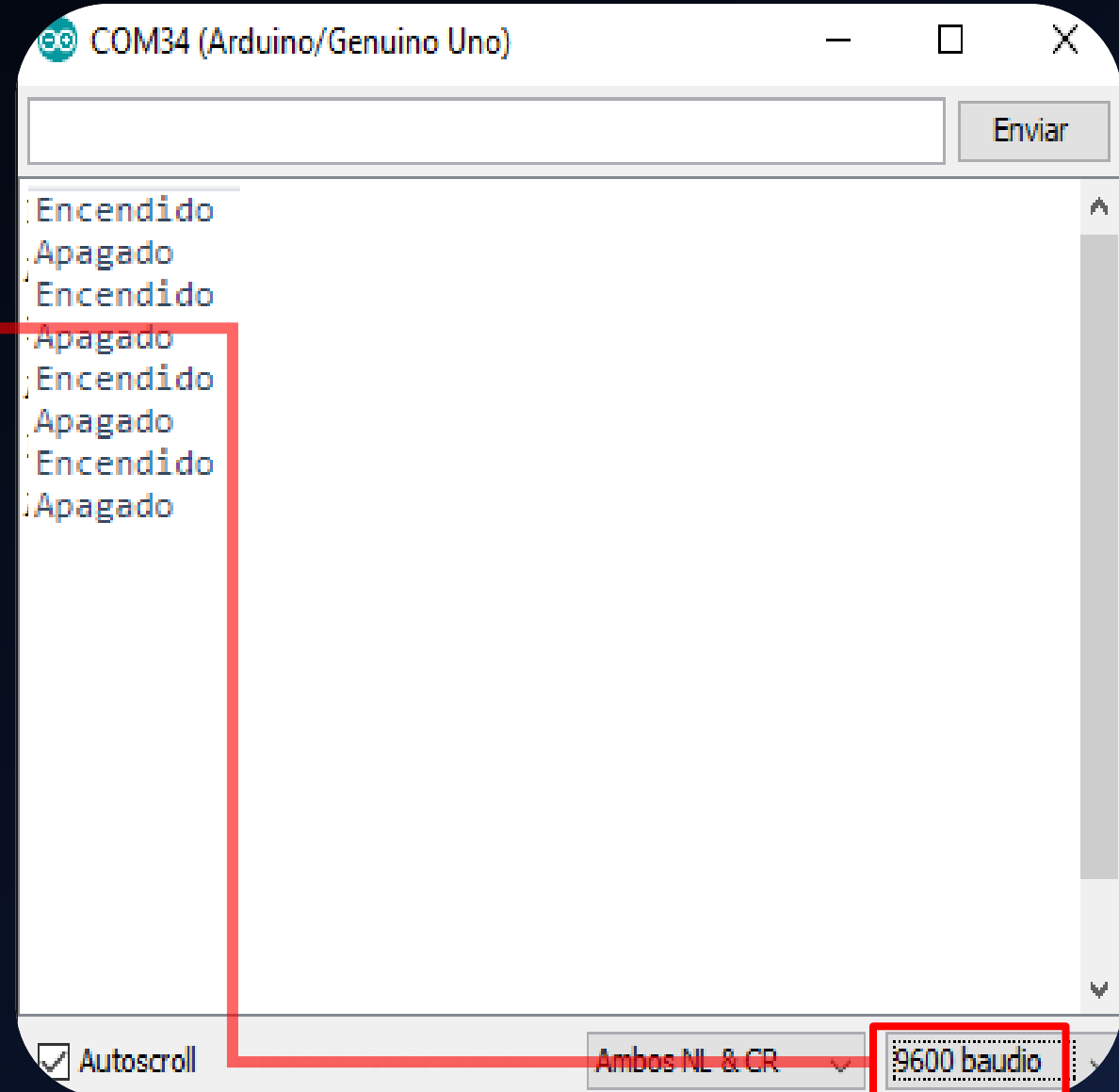
TUTOR: NAGIB LUIS VALLEJOS M.

# EJERCICIO 1 – SOL E IGUAL EN VELOCIDAD

S23-E1

```
1 int amarillo=4;
2 void setup() {
3   pinMode(amarillo, OUTPUT);
4   Serial.begin(9600);
5 }
6
7 void loop() {
8   Serial.println("Encendido");
9   digitalWrite(amarillo, 1);
10  delay(1000);
11  digitalWrite(amarillo, 0);
12  Serial.println("Apagado");
13  delay(1000);
14 }
```

=



TUTOR:NAGIB LUIS VALLEJOS M.

# UART – ENTRADA DE DATOS

Para poder realizar la recepción de datos, la velocidad del monitor serial debe ser el mismo del ingresado en el **Serial.begin()**.

Arduino dentro de su microcontrolador tiene integrado un buffer, el cual maneja una memoria volátil que es reiniciada, cada vez que dejamos de alimentar la placa.

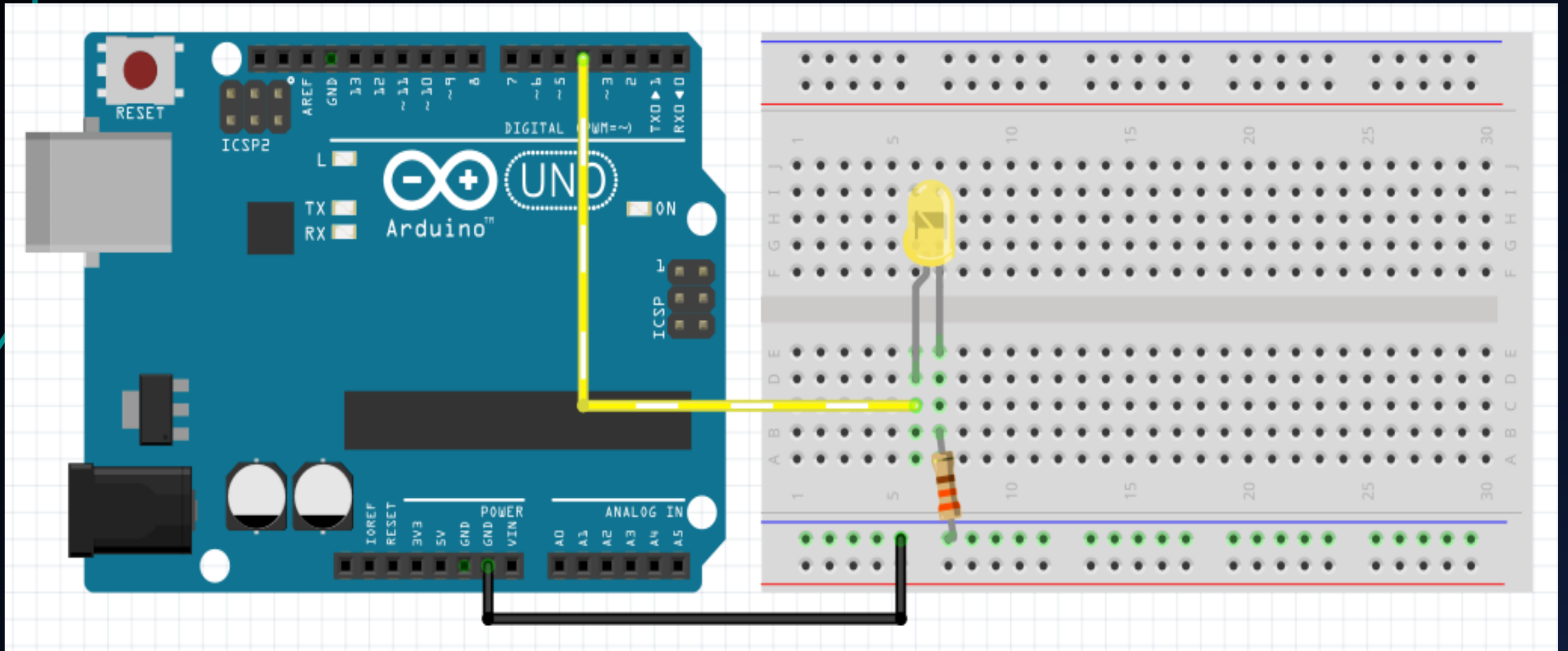
## **Serial.Available()**

Esta función nos permite registrar los datos que llegan del puerto serie y lo almacena en el buffer.



# EJERCICIO 2 – CIRCUITO

Realizar el encendido y apagado de un led a través del monitor serial. Enviar 'a' para encender el led y 'b' para apagarlo.



TUTOR: NAGIB LUIS VALLEJOS M.

# EJERCICIO 2 – SOLUCIÓN

S23-E2

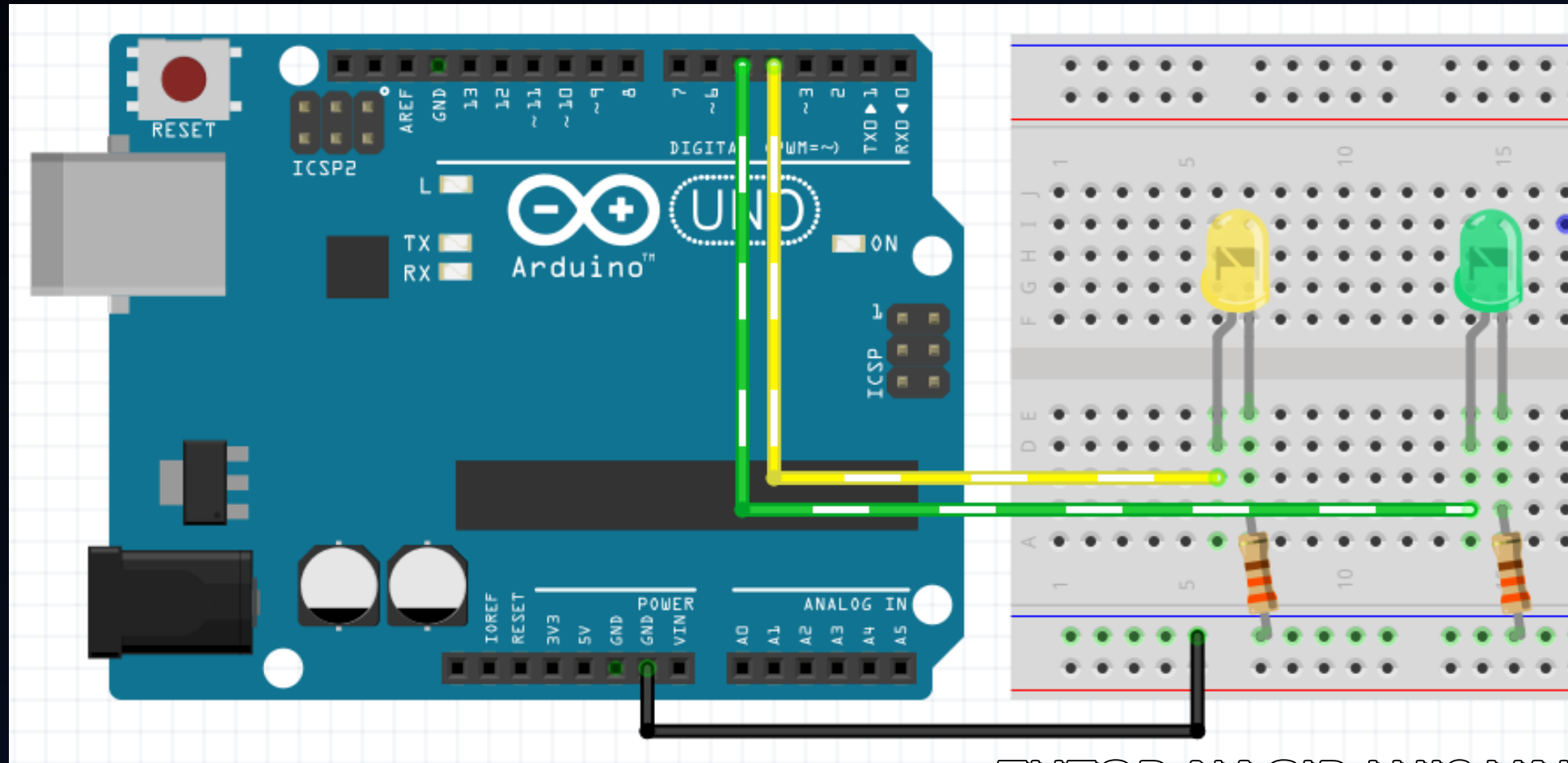
```
1 int amarillo=4;
2 char dato=' ';
3 void setup() {
4     pinMode(amarillo,OUTPUT);
5     Serial.begin(9600);
6 }
7 void loop() {
8     if(Serial.available()>0) { //almacenar datos en el buffer
9         dato=Serial.read();
10    }
11    if(dato=='a') { //dato==92
12        digitalWrite(amarillo,1);
13    }
14    else if(dato=='b') { //dato==93
15        digitalWrite(amarillo,0);
16    }
17 }
```

TUTOR:NAGIB LUIS VALLEJOS M.



# EJERCICIO 3 - CIRCUITO

Si envío 'a' enciende el led Amarillo y 'b' lo apaga. Si envío 'x' enciende el led Verde y 'z' lo apaga. Y si envío 'g' enciende ambos leds y 'h' los apaga



TUTOR:NAGIB LUIS VALLEJOS M.

# EJERCICIO 3 - SOLUCIÓN

Si envío 'a' enciende el led Amarillo y 'b' lo apaga. Si envío 'x' enciende el led Verde y 'z' lo apaga. Y si envío 'g' enciende ambos leds y 'h' los apaga

S23-E3

```
1 int amarillo=4,verde=5;
2 char dato=' ';
3 void setup() {
4   pinMode(amarillo,OUTPUT);
5   pinMode(verde,OUTPUT);
6   Serial.begin(9600);
7 }
8 void loop() {
9   if(Serial.available()>0){
10     dato=Serial.read();
11   }
12   if(dato=='a'){
13     digitalWrite(amarillo,1);
14   }
15   else if(dato=='b'){
16     digitalWrite(amarillo,0);
17   }
18   else if(dato=='x'){
19     digitalWrite(verde,1);
20   }
21   else if(dato=='z'){
22     digitalWrite(verde,0);
23   }
24   else if(dato=='g'){
25     digitalWrite(amarillo,1);
26     digitalWrite(verde,1);
27   }
28   else if(dato=='h'){
29     digitalWrite(amarillo,0);
30     digitalWrite(verde,0);
31   }
32 }
```

TUTOR:NAGIB LUIS VALLEJOS M.

# CONTACTOS

Suscríbete



(+591) 63096640



robotics.space.nv@gmail.com



fb.me/RoboticsSpaceNV



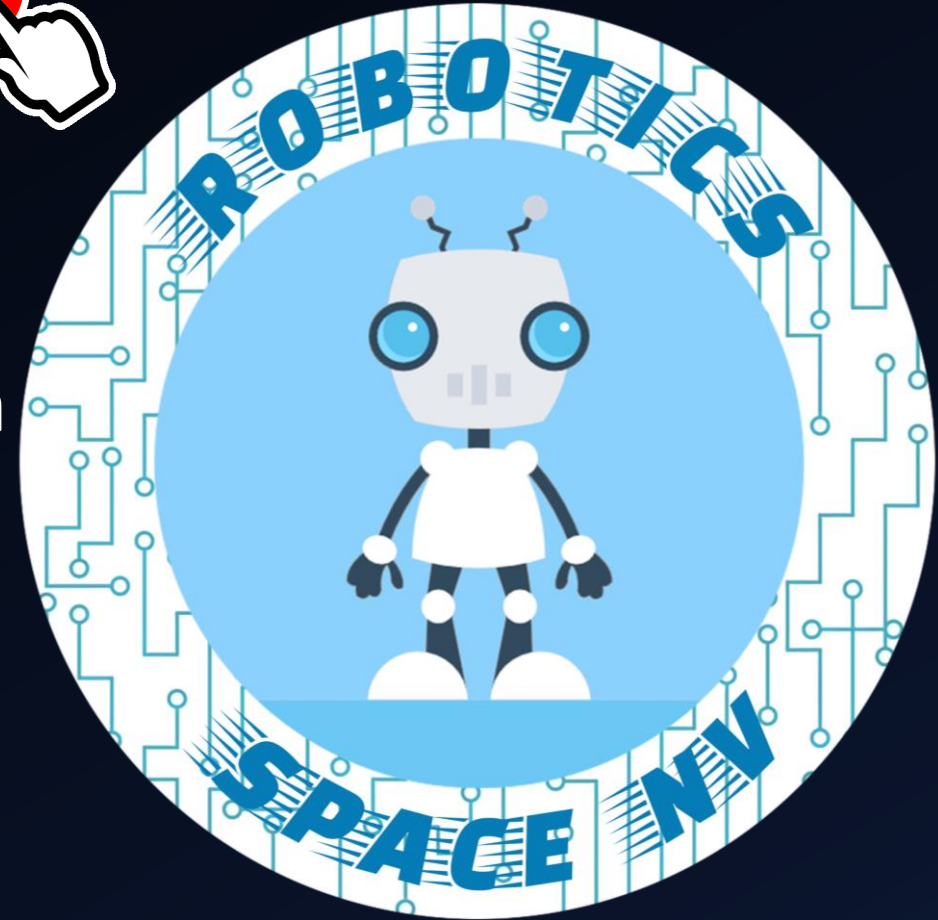
@NagibVallejos



Robotics Space NV



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



TUTOR:NAGIB LUIS VALLEJOS M.