

ESCUELA POLITÉCNICA NACIONAL

ESCUELA DE FORMACIÓN DE TECNÓLOGOS



DESARROLLO DE IoT

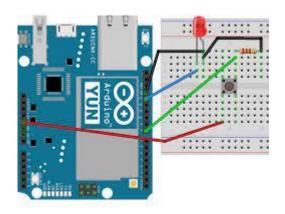
ASIGNATURA: Desarrollo de IoT PROFESOR: Ing. Vanessa Guevara

PERÍODO ACADÉMICO: 2024-A

TRABAJO EN CLASE

TÍTULO:

Adquisición de datos digitales



1. OBJETIVOS

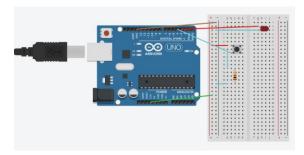
• Adquirir datos digitales.

2. MATERIALES:

- Placa Arduino
- Protoboard y cables de conexión.
- Resistencias
- Leds

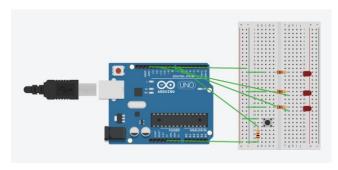
3. DESARROLLO

- 1. Realizar los siguientes programas en **TINKERCAD** (adjuntar capturas de pantalla del circuito y código):
- Encender un led usando un pulsador, de tal manera que al pulsarlo una vez active el led y
 pulsándolo otra vez desactive el led.



```
// C++ code
//variables
int pinPulsador = 4;
int estadoPulsador = 0;
    int pinLed = 3;
    int cambio = false;
    //Configuración
    void setup()
         pinMode(pinPulsador,INPUT); //entrada de datos
Serial.begin(9600);//comunicación serial (velocidad en vaudios)
         //configurar el pin del led como salida (output)
         pinMode(pinLed,OUTPUT);
18
19
    //Acciones a repetir.
    void loop()
         estadoPulsador = digitalRead(pinPulsador);
         Serial.println(estadoPulsador); //wer el estado del pulsador
26
27
      if(estadoPulsador==1 && cambio ==true){
         digitalWrite(pinLed, HIGH);
29
30
      if (estadoPulsador==1 && cambio ==false) {
  digitalWrite(pinLed, LOW);
33
34
       cambio = !cambio;
35
```

• Realizar el encendido de 3 LEDs de forma consecutiva; es decir, que se enciendan los LEDS en orden y de la misma manera se apaguen.



```
1 // Semáforo
2 // declarando parametros
3 int led verde = 13;
4 int led_amarillo = 12;
5 int led_rojo = 8;
6 const int BOTON = 7; // pin de entrada botón
7 int val = 0; // val se emplea para almacenar el estado
8 // del botón
9 int state = 0; // 0 LED apagado, mientras que 1 encendido
10 int old_val = 0; // almacena el antiguo valor de val// setup de parámetros
11 void setup() {
12 // se indica que cada pin es de salidad OUTPUT.
13 pinNode(led werde, OUTPUT);
14 pinNode(led amarillo, OUTPUT);
15 pinNode(led marillo, OUTPUT);
16 }
17 // lazo a ejecutar continuamente una vez cargado el código en el arduino
18 void loop() {
19 val digitalRead(BOTON); // lee el estado del Boton
10 // chequeer si el boton esta presionado o no
21 if ((val == HIGH) & (old_val == LOW)) {
22 state=1-state;
23 delay(10);
24 }
25 old_val = val; // val is now old, let's store it
26 if (state==1) {
27 digitalWrite(led_verde, HIGH); // encender LED verde
28 delay(5000); // mantener por 5 segundos
29 digitalWrite(led_verde, LOW); // apagar LED verde
30 digitalWrite(led_marillo, HIGH); // encender LED rojo
31 delay(1000); // mantener por 1 segundos
32 digitalWrite(led_marillo, HIGH); // encender LED rojo
33 digitalWrite(led_marillo, HIGH); // encender LED rojo
34 delay(1000);
35 digitalWrite(led_marillo, LOW); // apagar LED verde
36 digitalWrite(led_marillo, LOW); // apagar LED verde
36 digitalWrite(led_marillo, LOW); // encender LED rojo
36 delay(1000);
37 }
38 }
```

2. CONCLUSIONES

- El uso de un pulsador para controlar un LED permite entender la lógica básica de encendido y apagado en sistemas digitales.
- La secuencia de encendido de múltiples LEDs demuestra la importancia de la temporización y el control preciso en proyectos de IoT.
- La implementación práctica en Tinkercad facilita la visualización y comprensión de los circuitos antes de su montaje físico.

3. BIBLIOGRAFÍA

PRESENTACIÓN

Una vez culminada la tarea, subir con el nombre del archivo: Tarea_IoT_2024A_NApellido.