

# Programación Microcontrolador Arduino



## C3.2 Entradas digital

Arduino y entrada digital, utilizando un push button y una resistencia.



### Instrucciones

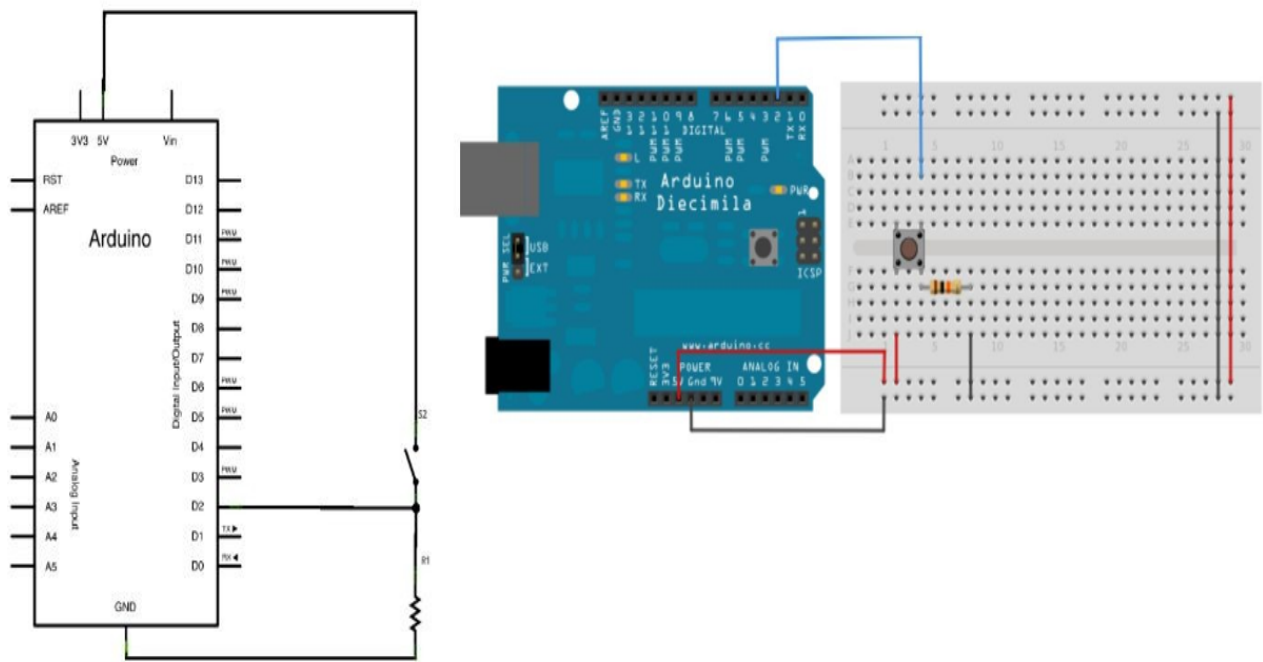
- De acuerdo con la información presentada por el asesor referente al tema, desarrollar lo que se indica dentro del apartado siguiente.
- Toda actividad o reto se deberá realizar utilizando el estilo **Markdown con extension .md** y el entorno de desarrollo VSCode, debiendo ser elaborado como un documento **single page**, es decir si el documento cuanta con imágenes, enlaces o cualquier documento externo debe ser accedido desde etiquetas y enlaces.
- Es requisito que el archivo .md contenga una etiqueta del enlace al repositorio de su documento en Github, por ejemplo **Enlace a mi GitHub**
- Al concluir el reto el reto se deberá subir a github el archivo .md creado.
- Desde el archivo .md se debe exportar un archivo .pdf con la nomenclatura **C3.2\_NombreAlumno\_Equipo.pdf**, el cual deberá subirse a classroom dentro de su apartado correspondiente, para que sirva como evidencia de su entrega; siendo esta plataforma **oficial** aquí se recibirá la calificación de su actividad por individual.
- Considerando que el archivo .pdf, fue obtenido desde archivo .md, ambos deben ser idénticos y mostrar el mismo contenido.
- Su repositorio ademas de que debe contar con un archivo **readme.md** dentro de su directorio raíz, con la información como datos del estudiante, equipo de trabajo, materia, carrera, datos del asesor, e incluso logotipo o imágenes, debe tener un apartado de contenidos o indice, los cuales realmente son ligas o **enlaces a sus documentos .md**, *evite utilizar texto* para indicar enlaces internos o externo.
- Se propone una estructura tal como esta indicada abajo, sin embargo puede utilizarse cualquier otra que le apoye para organizar su repositorio.

```
| readme.md
| | blog
| | | C3.1_TituloActividad.md
| | | C3.2_TituloActividad.md
| | | C3.3_TituloActividad.md
| | img
| | docs
| | | A3.1_TituloActividad.md
| | | A3.2_TituloActividad.md
```



### Desarrollo

1. Basado en el siguiente circuito, y utilizando uno de los simuladores propuestos, ensamblar lo que observa.



2. Analice y escriba el programa que se muestra a continuación.

```

DigitalReadSerial | Arduino 1.8.9
File Edit Sketch Tools Help

DigitalReadSerial
9 //
10
11 // digital pin 2 has a pushbutton attached to it. Give it a name:
12 int pushButton = 2;
13
14 // the setup routine runs once when you press reset:
15 void setup() {
16   // initialize serial communication at 9600 bits per second:
17   Serial.begin(9600);
18   // make the pushbutton's pin an input:
19   pinMode(pushButton, INPUT);
20 }
21
22 // the loop routine runs over and over again forever:
23 void loop() {
24   // read the input pin:
25   int buttonState = digitalRead(pushButton);
26   // print out the state of the button:
27   Serial.println(buttonState);
28   delay(1); // delay in between reads for stability
29 }

```

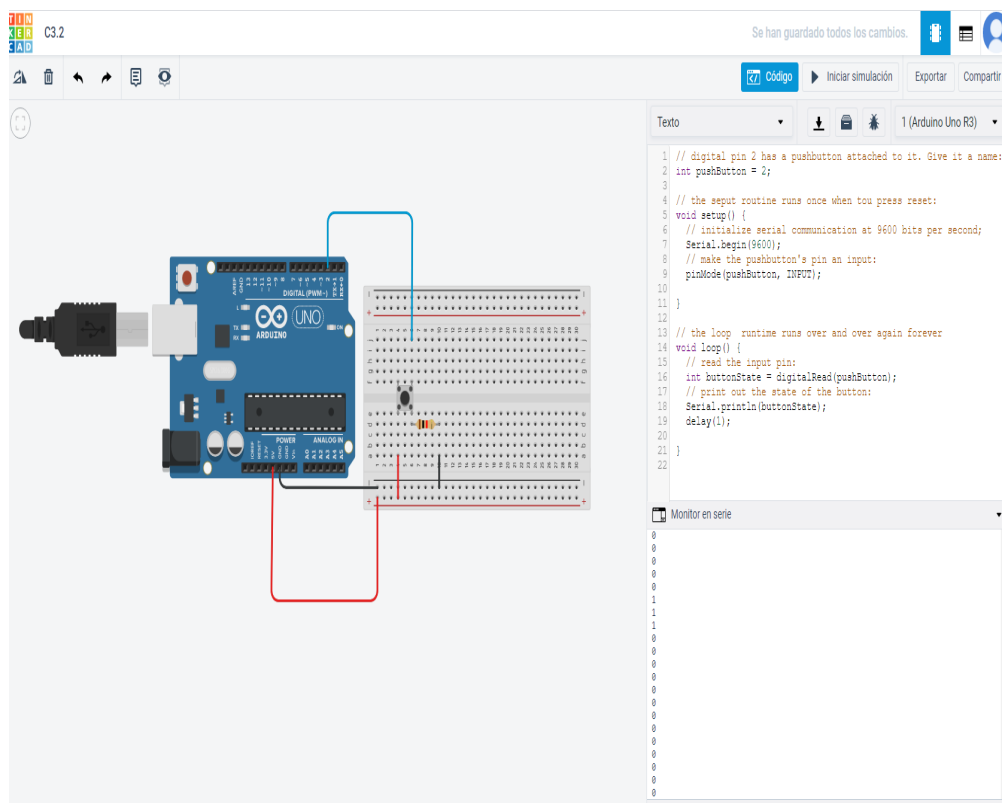
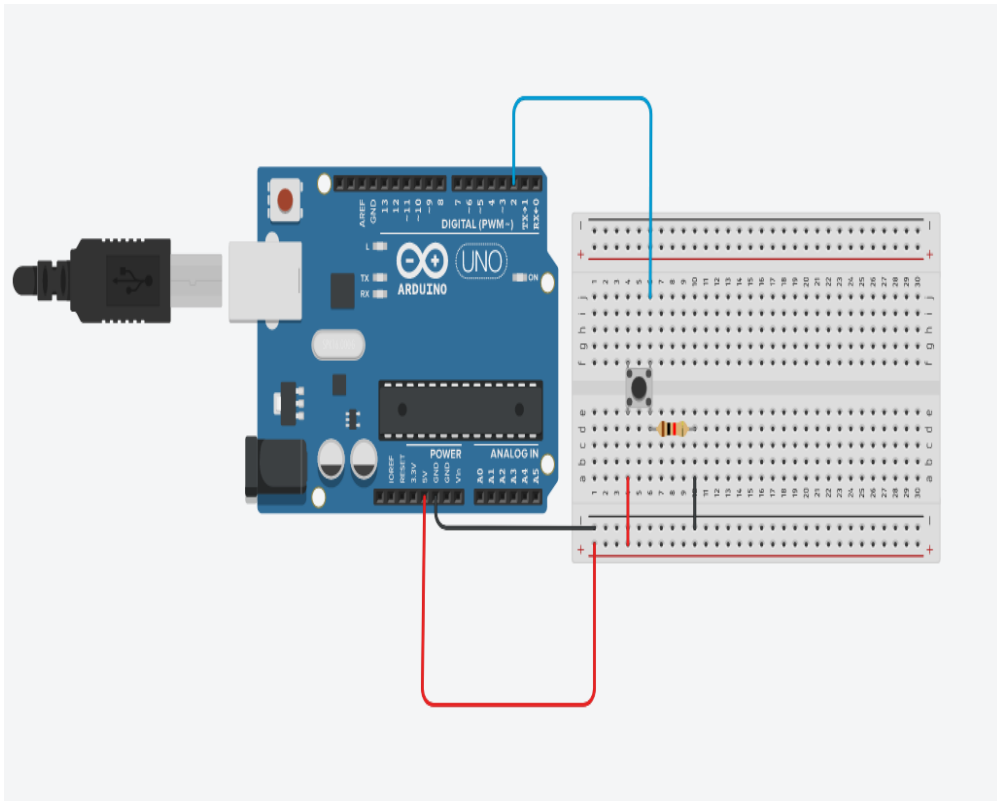
3. Explique el resultado observado.

Al inicio asignamos una variable para el pin 2 y lo configuramos como **INPUT**, así mismo asignamos al puerto serial una comunicación de 9600 bits por segundo. Ya en el bucle, primero asignamos el estado del pin 2 a una variable e imprimimos dicho estado, luego tenemos un descanso de 1 milisegundo y se repite el proceso.

Ahora, ¿Qué observamos?, que el botón siempre está recibiendo corriente desde la salida de 5V del Arduino, y cuando presionamos el botón, le pasa corriente a través de él, llegando al pin 2

del Arduino que hemos configurado como entrada, luego cambia su estado a **1** y lo imprime en el monitor en serie, mientras que no presionemos el botón, su estado será **0** y nos estará imprimiendo 0 en el puerto serial.

4. Inserte imágenes que considere como evidencias para demostrar el resultado obtenido.



```
1 // digital pin 2 has a pushbutton attached to it. Give it a name:
2 int pushButton = 2;
3
4 // the seput routine runs once when tou press reset:
5 void setup() {
6   // initialize serial communication at 9600 bits per second;
7   Serial.begin(9600);
8   // make the pushbutton's pin an input:
9   pinMode(pushButton, INPUT);
10
11 }
12
13 // the loop runtime runs over and over again forever
14 void loop() {
15   // read the input pin:
16   int buttonState = digitalRead(pushButton);
17   // print out the state of the button:
18   Serial.println(buttonState);
19   delay(1);
20 }
21
22
```

Monitor en serie

0  
0  
0  
0  
0  
1  
1  
1  
0  
0  
0  
0  
0  
0  
0  
0  
0  
0  
0

### :bomb: Rubrica

Criterios	Descripción	Puntaje
Instrucciones	Se cumple con cada uno de los puntos indicados dentro del apartado Instrucciones?	20
Desarrollo	Se respondió a cada uno de los puntos solicitados dentro del desarrollo de la actividad?	80

 [Ir a readme](#)

 [Ver en repositorio](#)