	Computación	Docente: Diego Quisi Peralta
	Programación Aplicada	Período Lectivo: Septiembre 2020 – Febrero 2021

		FORMATO DE GUÍA DE PRÁCTICA DE LABORATORIO / TALLERES / CENTROS DE SIMULACIÓN – PARA DOCENTES	
CARRERA: COMPUTACIÓN/INGENIERÍA DE SISTEMAS		ASIGNATURA: PROGRAMACIÓN APLICADA	
NRO. PROYECTO:	1.1	TÍTULO PROYECTO: Practica de Arduino Desarrollo e implementación de un entorno de simulación Online para placas Arduino y electronica basica.	
OBJETIVO: Reforzar los conocimientos adquiridos en clase sobre Arduino.			
INSTRUCCIONES:		1. Revisar el contenido teórico y practico del tema.	
		2. Profundizar los conocimientos revisando los libros guías, los enlaces contenidos en los objetos de aprendizaje Java y la documentación disponible en fuentes académicas en línea de Arduino.	
		3. Crear una cuenta dentro de la herramienta Online https://www.tinkercad.com para simular circuitos electricos.	
		4. Revisar los siguientes videos que le ayudaran para realizar la tarea: - https://www.youtube.com/watch?v=r25dG32IWSU (Video de Electrónica Básica) - https://www.youtube.com/watch?v=hZmSG-IALAM (Video de Arduino Básico)	
		5. Revisar el ejemplo subido al AVAC del prender un led dentro del simulador TinkerCad y cargar en la herramienta para ver la simulación (PrenderApagarLed.brd). 6. Subir el informe de la practica en formato PDF y los archivos al GitPersonal.	
		Fecha de Entrega: 31 de Enero 2021	
ACTIVIDADES POR DESARROLLAR			

1. Investigue, diseñe y desarrolle e implemente tres sistema de simulación electrónica de Arduino dentro de la herramienta online Thincad.

DEFINICIÓN DEL PROBLEMA:

TinkerCad es un software gratuito para el diseño 3D desarrollado por Autodesk. En su apartado circuits ofrece un simulador online de Arduino bastante completo y facil de utilizar.

TinkerCad ofrece bastantes componentes para armar nuestros esquemas y circuitos, y muchos de ellos se pueden configurar (como por ejemplo las resistencias y los diodos) y manipular en tiempo real (potenciómetros, botones, etc.).

La programación en TinkerCad se puede realizar en modo código y en modo bloques, y también tenemos disponible una pantalla dividida donde vemos los dos modos simultáneamente.

Al compilar el código, si hay algún error es marcado por el depurador. Una vez que tenemos el código arduino listo y la simulación funciona sin errores podemos descargar el archivo .ino para subirlo a nuestro arduino. El software nos provee de un monitor serie con plotter serial incluido similar al IDE de Arduino. TinkerCad es una plataforma ideal para quienes están aprendiendo Arduino y programación. Es muy intuitiva y de fácil manejo, gratuita y online.

En base a ello se propone resolver tres problemas electrónicos:

1. Generar un autofantastico que se prenda y se apague desde un pulsante.
2. Generar una lampara de ciudad, es decir que se prenda cuando es noche y se apague cuando ya exista luz para esto deben utilizar un LDR y un LED.
3. Finalmente, controlar un servomotor con un potenciómetro el grado de giro.

RESULTADO(S) OBTENIDO(S):

- Interpreta de forma correcta la programación en Arduino.
- Identifica correctamente qué herramientas de electronicas se pueden aplicar.

CONCLUSIONES:

- Los estudiantes implementan soluciones de hardware en sistemas.
- Los estudiantes estan en la capacidad de implementar sistemas electronicos en Arduino.

RECOMENDACIONES:


- Revisar la información proporcionada por el docente previo a la práctica.
- Haber asistido a las sesiones de clase.
- **Consultar con el docente las dudas que puedan surgir al momento de realizar la práctica.**


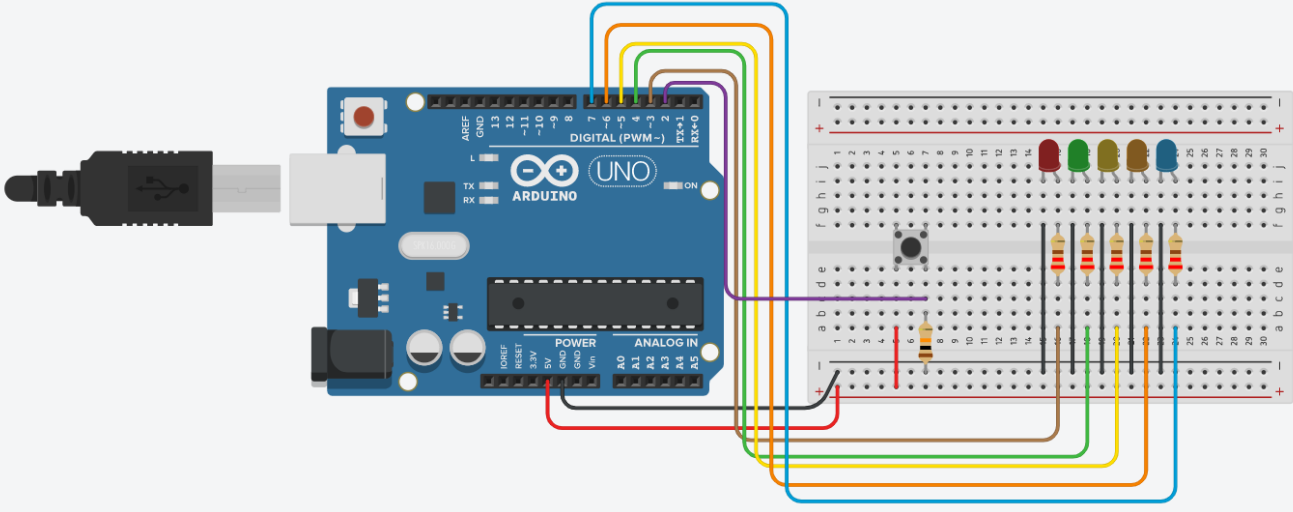
BIBLIOGRAFIA:

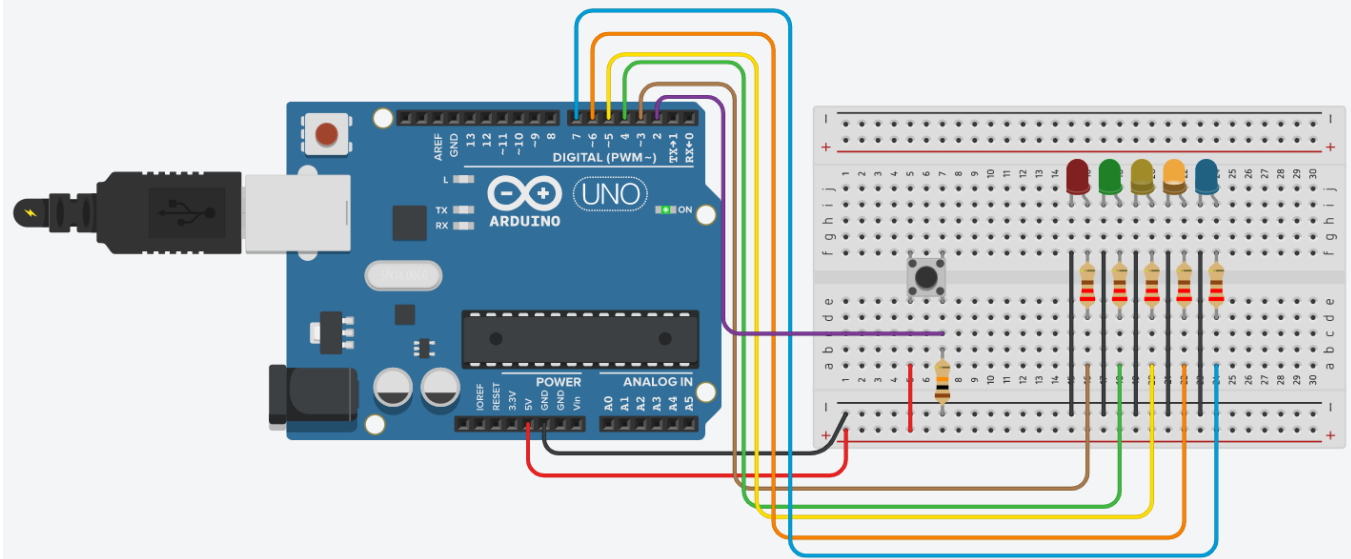
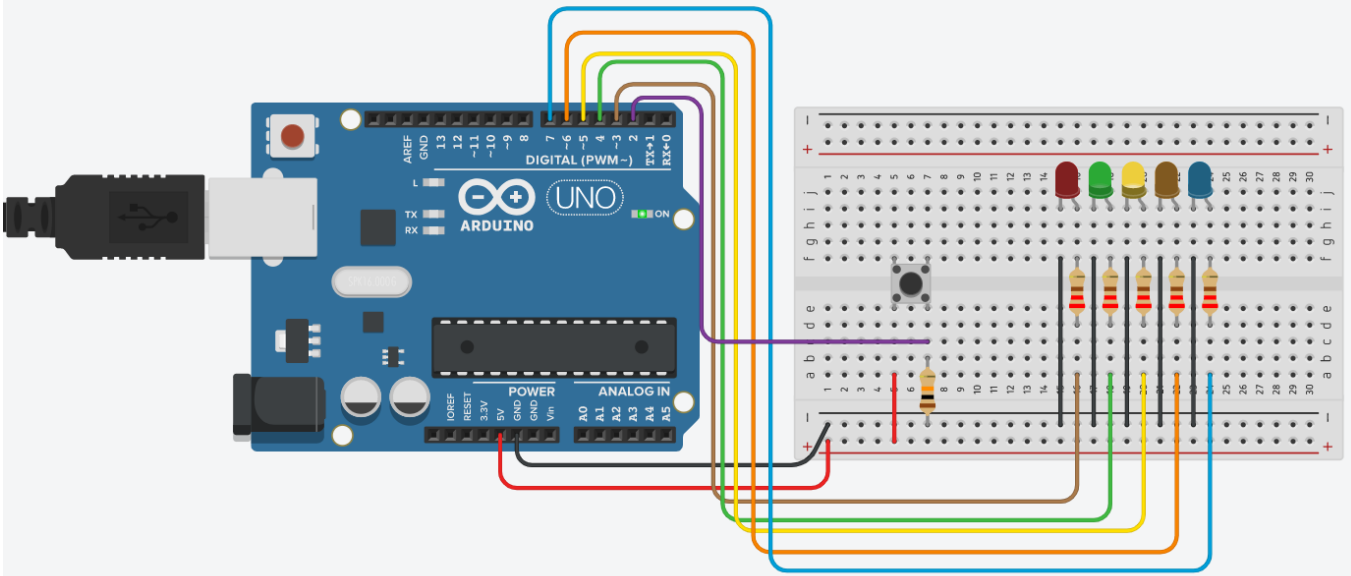
[1]: <https://www.ups.edu.ec/evento?calendarBookingId=98892>

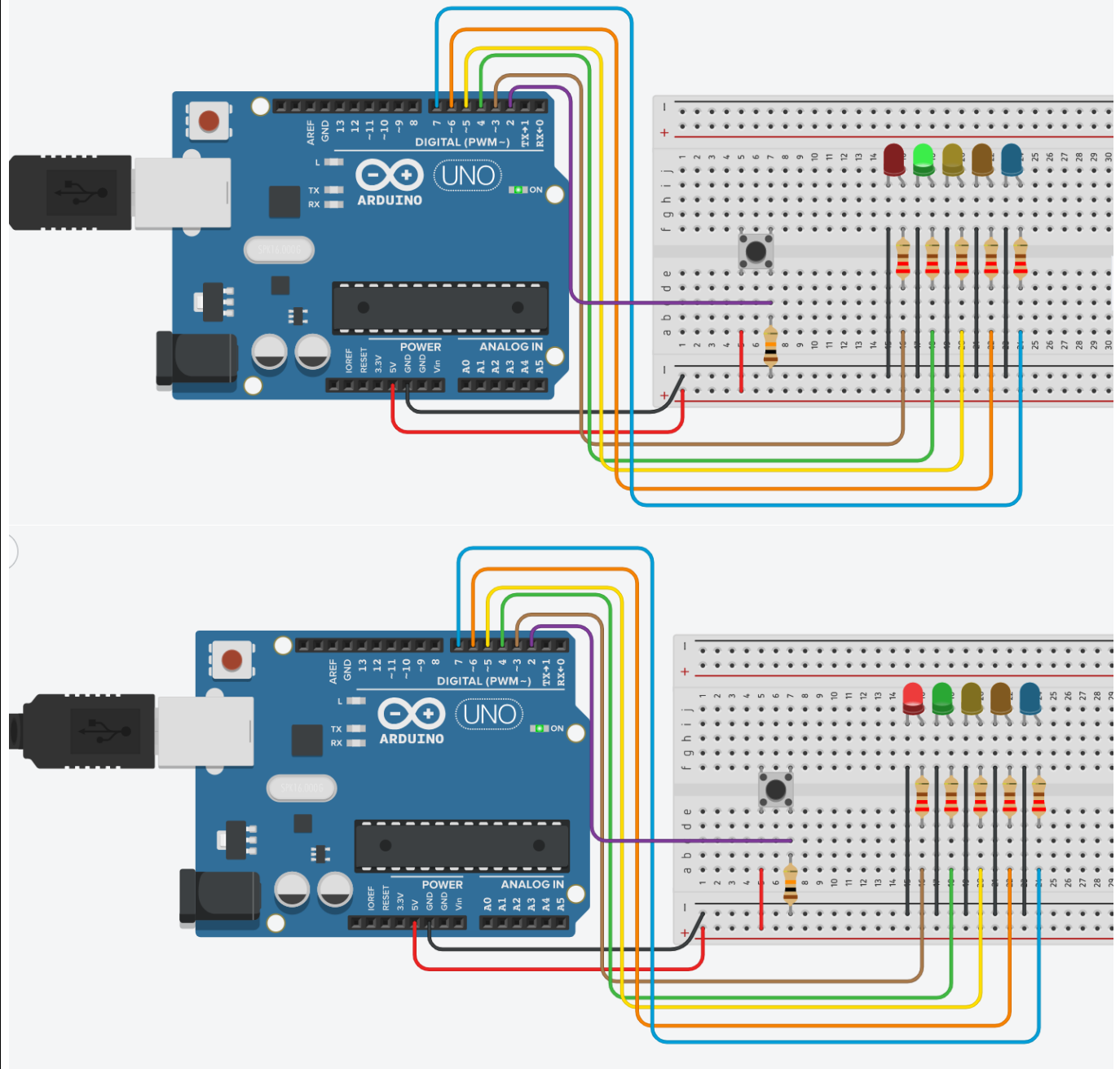
Docente / Técnico Docente: Ing. Diego Quisi Peralta Msc.

Firma: _____

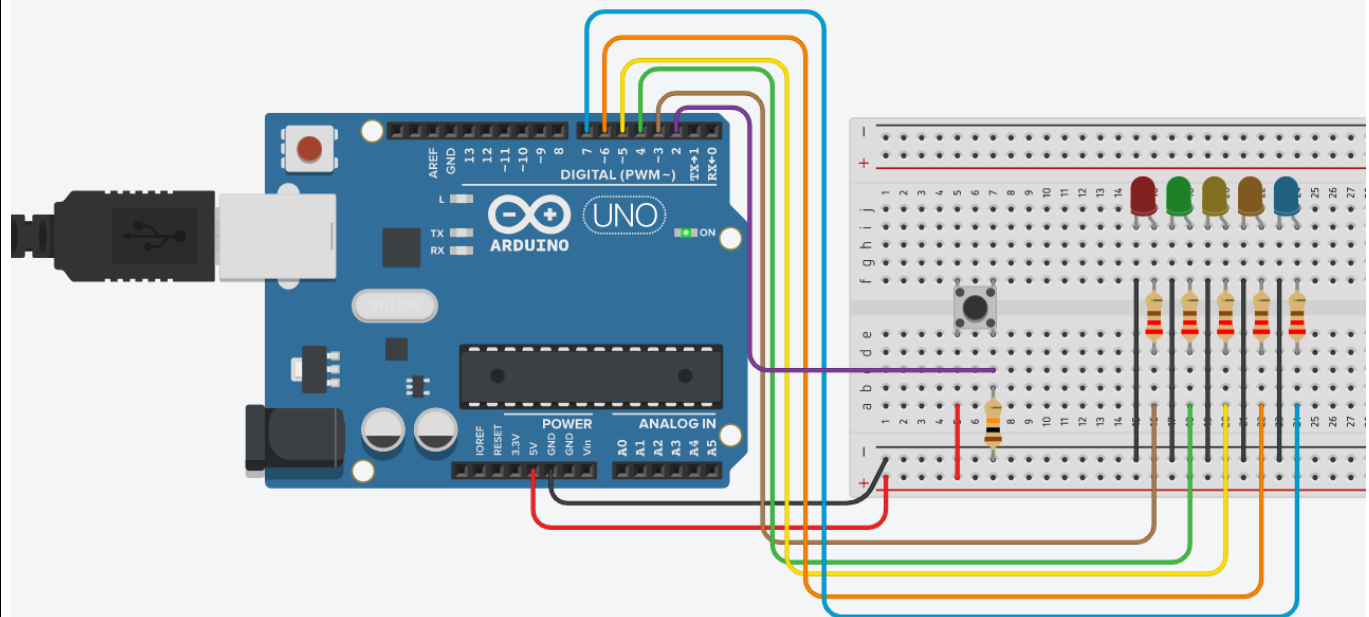
	Computación	Docente: Diego Quisi Peralta
	Programación Aplicada	Período Lectivo: Septiembre 2020 – Febrero 2021

		FORMATO DE INFORME DE PRÁCTICA DE LABORATORIO / TALLERES / CENTROS DE SIMULACIÓN – PARA ESTUDIANTES	
CARRERA: COMPUTACIÓN/INGENIERÍA DE SISTEMAS		ASIGNATURA: PROGRAMACIÓN APLICADA	
NRO. PROYECTO:	1.1	TÍTULO PROYECTO: Practica de Arduino Desarrollo e implementación de un entorno de simulación Online para placas Arduino y electronica basica.	
OBJETIVO: Reforzar los conocimientos adquiridos en clase sobre Arduino.			
ACTIVIDADES DESARROLLADAS			
1. Generar un autofantástico que se prenda y se apague desde un pulsante.			
			
Conexión del Arduino al protoboard y en el proto se encuentra un pulsante 5 leds de diferente color Y 6 resistencias			

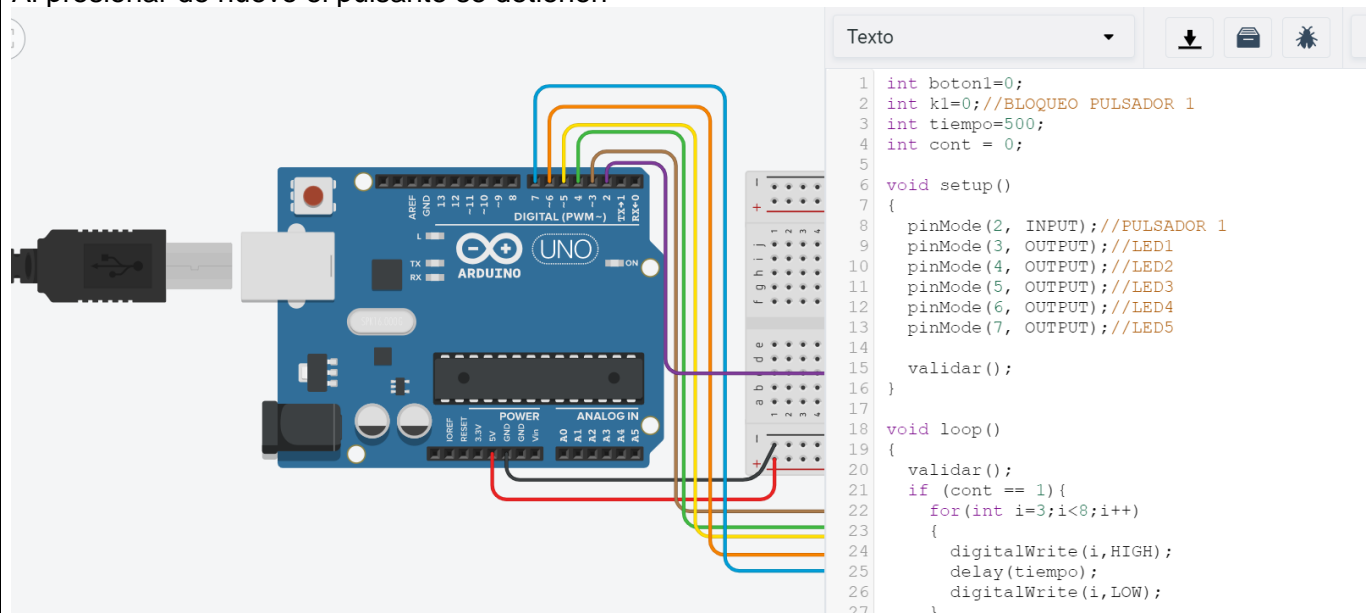




Comprobamos que los leds se encienden de forma ascendente y descendente al presionar el pulsante



Al presionar de nuevo el pulsante se detienen



```
int boton1=0;
```

```
int k1=0;//BLOQUEO PULSADOR 1
```

```
int tiempo=500;
```

```
int cont = 0;
```

```
void setup()
```

```
{
```

```
    pinMode(2, INPUT);//PULSADOR 1
```

```
    pinMode(3, OUTPUT);//LED1
```

```
pinMode(4, OUTPUT);//LED2
pinMode(5, OUTPUT);//LED3
pinMode(6, OUTPUT);//LED4
pinMode(7, OUTPUT);//LED5
validar();
}
void loop()
{
  validar();
  if (cont == 1){
    for(int i=3;i<8;i++)
    {
      digitalWrite(i,HIGH);
      delay(tiempo);
      digitalWrite(i,LOW);
    }
    for(int i=7;i>2;i--)
    {
      digitalWrite(i,HIGH);
      delay(tiempo);
      digitalWrite(i,LOW);
    }
  }
}
void validar(){
  if (digitalRead(2)){
    cont = cont + 1;
    if(cont>1){
      cont = 0;
    }
  }
}
```

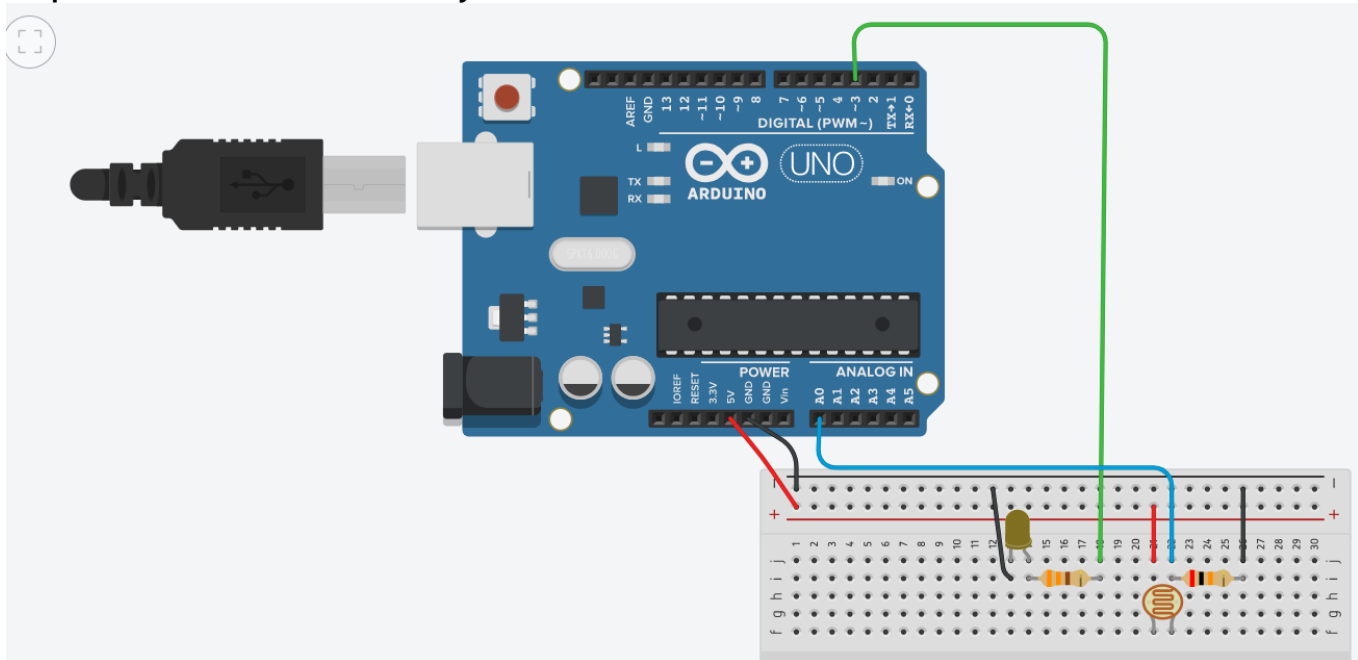
```

}
}

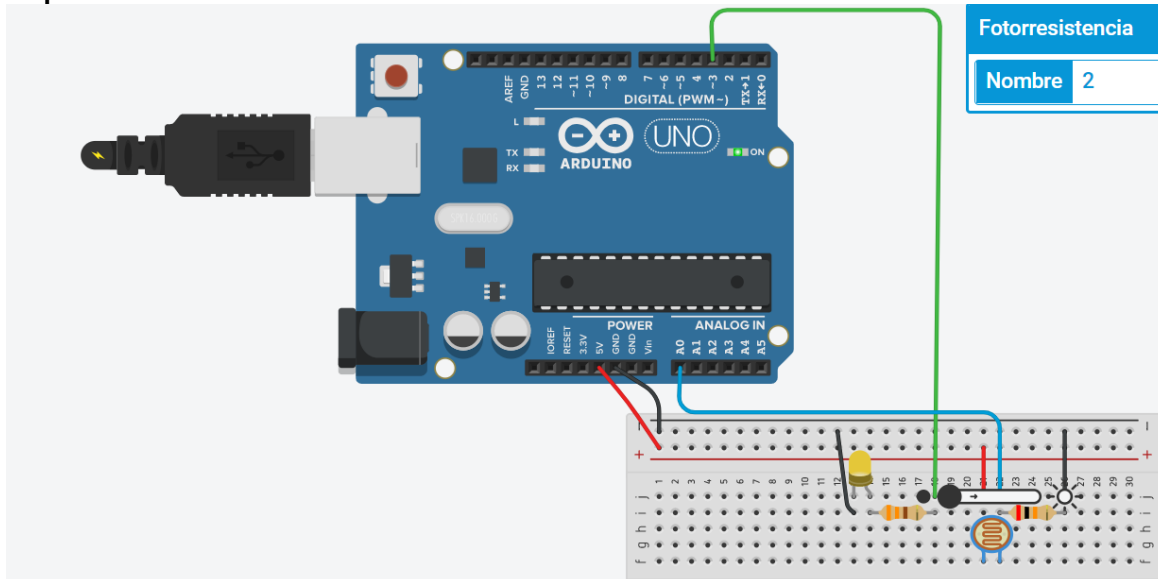
```

En el método ponemos que la entrada va a ser el pulsante y la salida va a ser los leds en los puertos 2,3,4,5,6,7 al ejecutar primero validamos si no está presionado el pulsante y entonces mandamos a `if` dentro del mismo tenemos un `for` que hará que primero se prenda el led espere un tiempo y se apague y después este otro `for` que nos ayuda hacer en encendido en descendente.

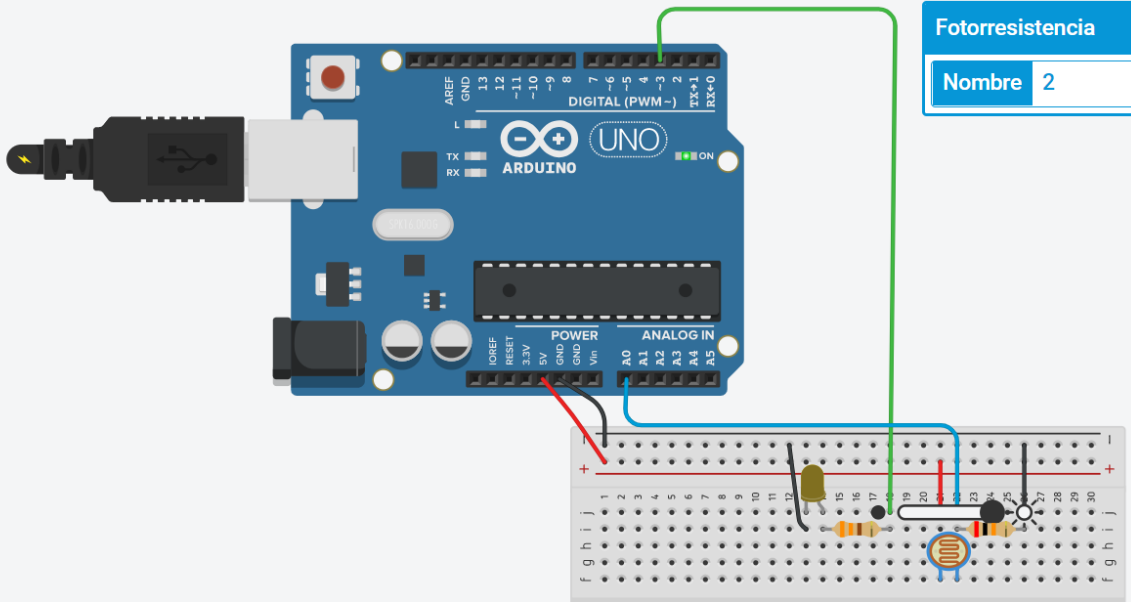
2. Generar una lámpara de ciudad, es decir que se prenda cuando es noche y se apague cuando ya exista luz para esto deben utilizar un LDR y un LED.



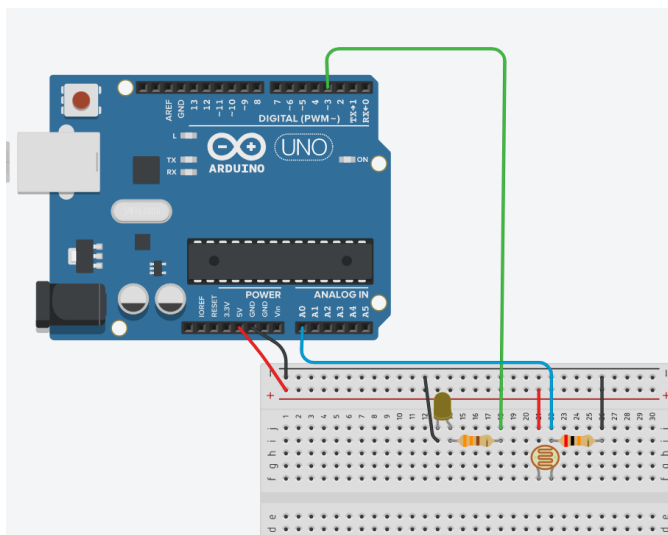
Conexión del Arduino al protoboard y a los elementos en el mismo en este caso tenemos un led dos resistencias y un LDR o sensor de luz que nos ayudara a ver si está con luz no se prendera y si no está se prendera



Corremos la simulación y vemos que no hay luz por lo tanto se enciende el led



Le damos luz al LDR y vemos como se apaga el led



Texto

```

1  const int fotopin = A0;
2  int valorSensor = 0;
3  const int led = 3;
4  int valorMapeado = 0;
5
6
7  void setup()
8  {
9      pinMode(fotopin, INPUT);
10     pinMode(led, OUTPUT);
11     Serial.begin(9600);
12 }
13
14 void loop()
15 {
16     valorSensor = analogRead(fotopin);
17     valorMapeado = map(valorSensor, 998, 102, 0, 255);
18
19     analogWrite(led, valorMapeado);
20
21     Serial.print("Valor del sensor ");
22     Serial.println(valorSensor);
23
24     Serial.print("Valor Mapeado ");
25     Serial.println(valorMapeado);
26
27 }
```

const int fotopin = A0;

int valorSensor = 0;

const int led = 3;

int valorMapeado = 0;

void setup()

```

{
    pinMode(fotopin, INPUT);
    pinMode(led, OUTPUT);
    Serial.begin(9600);
}
```

void loop()

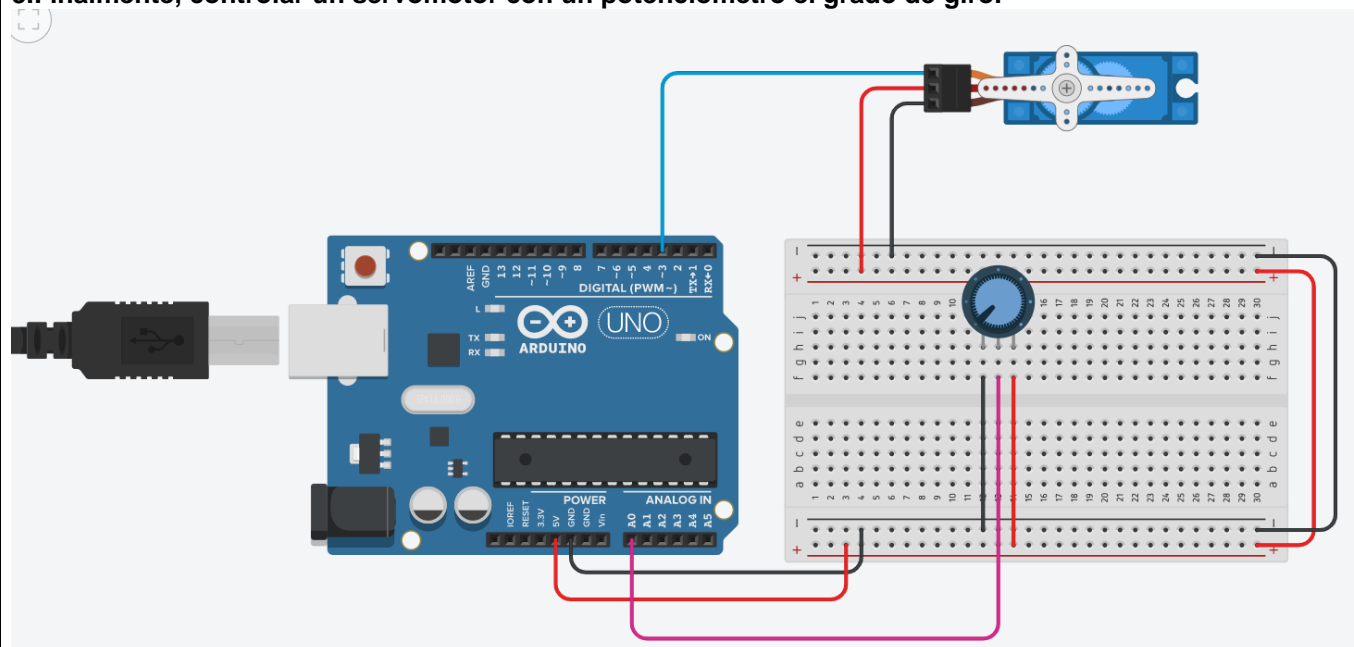
```

{
  valorSensor = analogRead(fotopin);
  valorMapeado = map(valorSensor,998,102,0,255);
  analogWrite(led, valorMapeado);
  Serial.print("Valor del sensor ");
  Serial.println(valorSensor);
  Serial.print("Valor Mapeado ");
  Serial.println(valorMapeado);
}

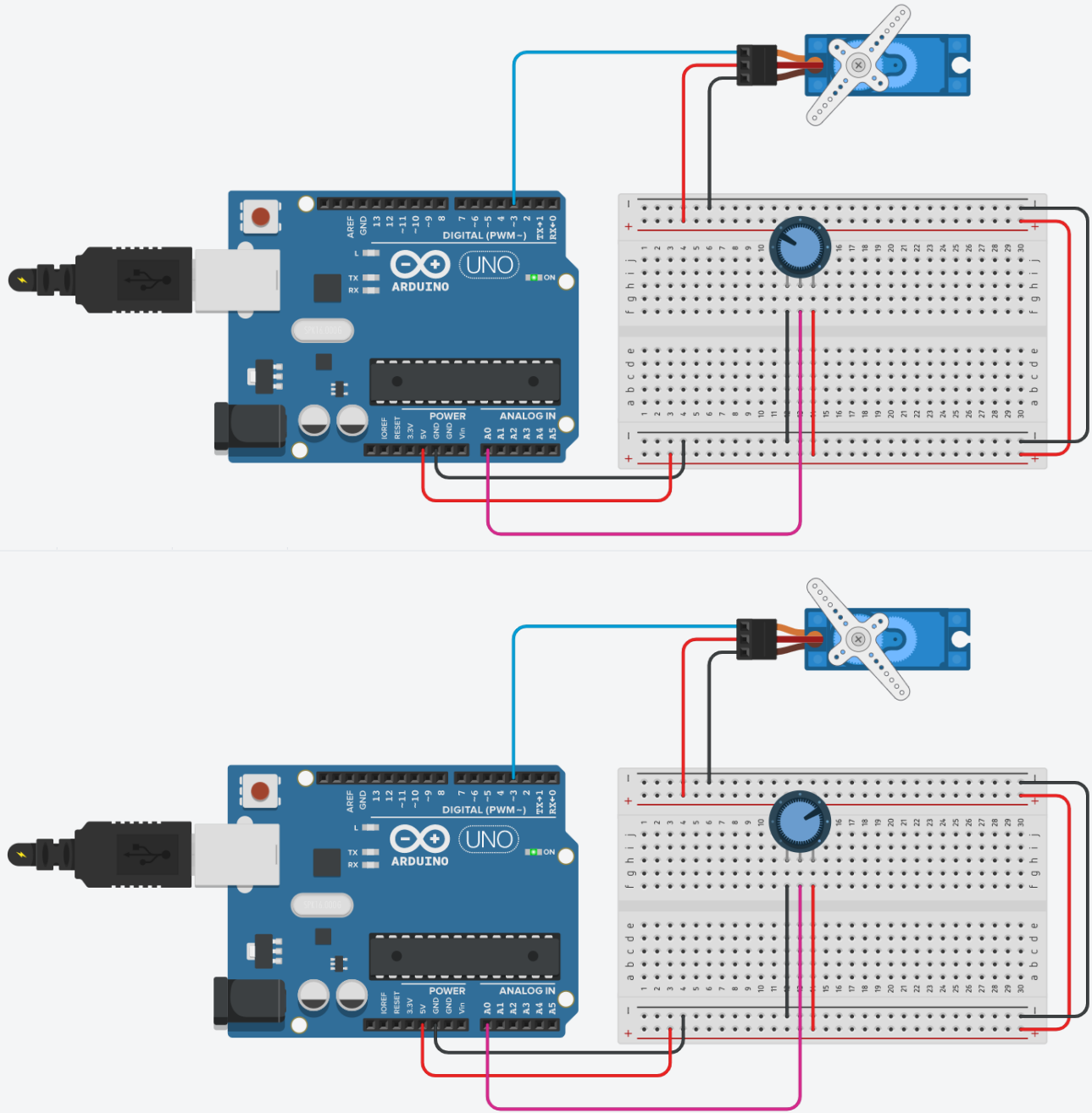
```

Código para que funcione el programa con el Arduino declaramos los puertos de entrada en este caso el LDR se encuentra en el A0 y el led esta en el 3 después en el método declaramos que el LDR será la entrada y el led será la salida y finalmente para saber cuando se apaga el led cuando tenga luz mapeamos los valores dándole el rango de la entra del LDR y según eso se prendera o apagara el foco.

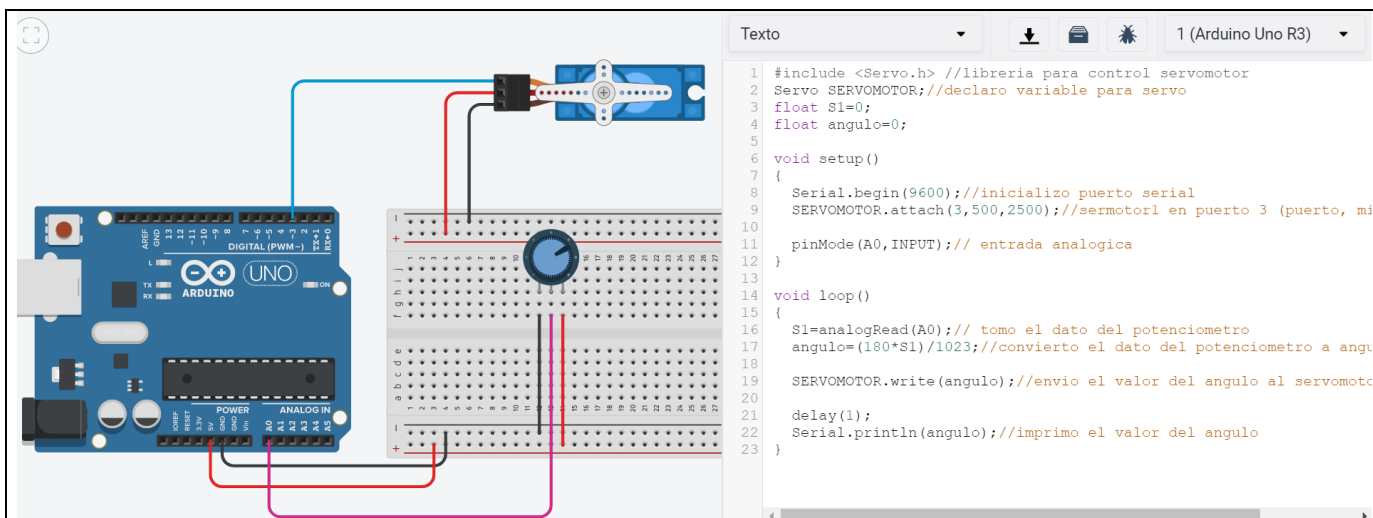
3.Finalmente, controlar un servomotor con un potenciómetro el grado de giro.



Conexión servomotor potenciómetro y el Arduino en el protoboard



Movimiento angular del servomotor al mover el potenciómetro



```
#include <Servo.h> //libreria para control servomotor
```

```
Servo SERVOMOTOR; //declaro variable para servo
```

```
float S1=0;
```

```
float angulo=0;
```

```
void setup()
```

```
{
```

```
  Serial.begin(9600); //inicializo puerto serial
```

```
  SERVOMOTOR.attach(3,500,2500); //servomotor1 en puerto 3 (puerto, minimo, maximo)
```

```
  pinMode(A0,INPUT); // entrada analogica
```

```
}
```

```
void loop()
```

```
{
```

```
  S1=analogRead(A0); // tomo el dato del potenciómetro
```

```
  angulo=(180*S1)/1023; //convierto el dato del potenciómetro a angulo
```


```
  SERVOMOTOR.write(angulo); //envio el valor del angulo al servomotor
```

```
  delay(1);
```

```
  Serial.println(angulo); //imprimo el valor del angulo
```

```
}
```

Código para hacer que se mueva el servomotor de acuerdo con los ángulos del potenciómetro primero importamos la librería que controla el servomotor después declaramos una variable y en el método ponemos que la entrada del puerto A0 en donde está el potenciómetro será la entrada y en el otro método leemos lo que nos da el puerto A0 y lo convertimos en ángulos para pasarle al servomotor.

	Computación	Docente: Diego Quisi Peralta
	Programacion Aplicada	Período Lectivo: Septiembre 2020 – Febrero 2021

RESULTADO(S) OBTENIDO(S):

Como resultados obtenidos podemos encontrar que podemos programar en Arduino para así realizar diferentes acciones desde encendido de leds a movimiento de servomotores

CONCLUSIONES:

En conclusión, programar en un Arduino no es difícil y nos ayuda a ver físicamente lo que estamos haciendo para que funcione de una manera correcta.

RECOMENDACIONES:

No hay Recomendaciones

Estudiantes: Romel Ávila

Firma:

