
	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 electronics 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>

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

Firma: _____

CARRERA: Computación

ASIGNATURA: Programación Aplicada

NRO. PRÁCTICA:

1.1

TÍTULO PRÁCTICA: Arduino

OBJETIVO ALCANZADO:

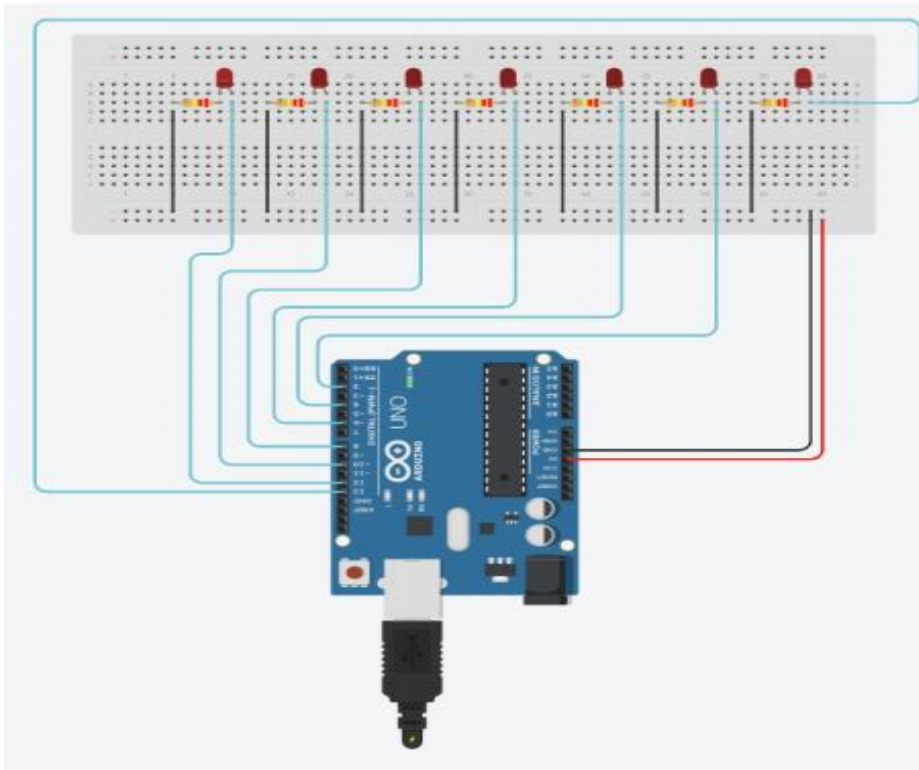
Reforzar los conocimientos adquiridos en clase sobre Arduino.

ACTIVIDADES DESARROLLADAS

1. Generar un autofantástico que se prenda y se apague desde un pulsante.

```
void setup()
{
    pinMode(12, OUTPUT);
    pinMode(10, OUTPUT);
    pinMode(8, OUTPUT);
    pinMode(6, OUTPUT);
    pinMode(4, OUTPUT);
    pinMode(2, OUTPUT);
    pinMode(13, OUTPUT);
}

void loop()
{
    digitalWrite(12, HIGH);
    delay(200); // Wait for 200 millisecond(s)
    digitalWrite(12, LOW);
    digitalWrite(10, HIGH);
    delay(200); // Wait for 200 millisecond(s)
    digitalWrite(10, LOW);
    digitalWrite(8, HIGH);
    delay(200); // Wait for 200 millisecond(s)
    digitalWrite(8, LOW);
    digitalWrite(6, HIGH);
    delay(200); // Wait for 200 millisecond(s)
    digitalWrite(6, LOW);
    digitalWrite(4, HIGH);
    delay(200); // Wait for 200 millisecond(s)
    digitalWrite(4, LOW);
    digitalWrite(2, HIGH);
    delay(200); // Wait for 200 millisecond(s)
    digitalWrite(2, LOW);
    digitalWrite(13, HIGH);
    delay(200); // Wait for 200 millisecond(s)
    digitalWrite(13, LOW);
}
```



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.

```

1 //Código realizado por: Anahí Cabrera
2 //Computación- Programación Aplicada
3 const int LDR=A0; // Se le asigna el Pin Analógico 0 al sensor
4 int valorSensor=0;
5 const int LED=3; //Pin numero 3 para el LED
6 int valorMapeado=0; //Valores que dará el sensor
7 void setup() {
8     pinMode (LDR, INPUT); //Asignacion del LDR como entrada
9     pinMode(LED, OUTPUT); //Asignacion del LED como salida
10    Serial.begin(9600);
11
12 }
13
14 void loop() {
15     valorSensor= analogRead(LDR);
16     valorMapeado=map(valorSensor,975,50,0,255); //(maximo de luz,minimo luz,LED encendido, LED apagado)
17     analogWrite(LED, valorMapeado);
18     Serial.print("Valor del Sensor: ");
19     Serial.println(valorSensor);
20     Serial.print("Valor del Sensor Mapeado:");
21     Serial.print(valorMapeado);
22
23 }
    
```

Simulator time: 00:00:00.320

Photoresistor
Name 2

Photoresistor
Name 2

```

1 //Código realizado por: Anah
2 //Computación- Programación A
3 const int LDR=A0; // Se le a
4 int valorSensor=0;
5 const int LED=3; //Pin numer
6 int valorMapeado=0; //Valore
7 void setup() {
8   pinMode (LDR, INPUT); //As
9   pinMode(LED, OUTPUT); //As
10  Serial.begin(9600);
11
12 }
13
14 void loop() {
15   valorSensor= analogRead(LD
16   valorMapeado=map(valorSens
17   analogWrite(LED, valorMape
18   Serial.print("Valor del Se
19   Serial.println(valorSensor
20   Serial.print("Valor del Se
21   Serial.print(valorMapeado)
22
23

```

Text

```

1 //Código realizado por: An
2 //Computación- Programación
3 const int LDR=A0; // Se le
4 int valorSensor=0;
5 const int LED=3; //Pin num
6 int valorMapeado=0; //Valo
7 void setup() {
8   pinMode (LDR, INPUT); //
9   pinMode(LED, OUTPUT); //
10  Serial.begin(9600);
11
12 }
13
14 void loop() {
15   valorSensor= analogRead(
16   valorMapeado=map(valorSe
17   analogWrite(LED, valorMa
18   Serial.print("Valor del
19   Serial.println(valorSens
20   Serial.print("Valor del
21   Serial.print(valorMapead
22
23 }

```

Text

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

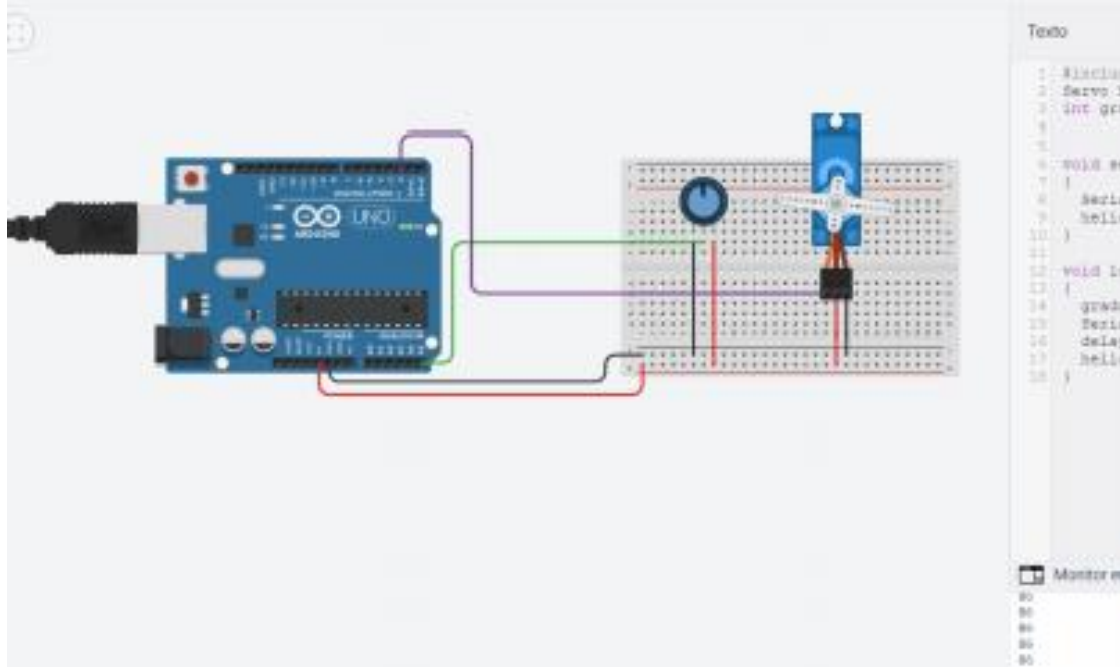
```

#include <Servo.h>
Servo helice;
int grados;

void setup()
{
  Serial.begin(9600);
  helice.attach(2);
}

void loop()
{
  grados = map(analogRead(A5), 0 ,1023,0 , 180);
  Serial.println(grados);
  delay(500);
  helice.write(grados);
}

```



RESULTADO(S) OBTENIDO(S):

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

CONCLUSIONES:

- Los estudiantes implementan soluciones de hardware en sistemas.
- Los estudiantes están en la capacidad de implementar sistemas electrónicos en Arduino.

RECOMENDACIONES:

Revisar conceptos básicos de electrónica.

Nombre de estudiante: Edith Anahí Cabrera Bermeo

Firma de estudiante: