	<b>Computación</b>	<b>Docente: Diego Quisi Peralta</b>
	Programación Aplicada	<b>Período Lectivo:</b> Septiembre 2020 – Febrero 2021

		<b>FORMATO DE GUÍA DE PRÁCTICA DE LABORATORIO / TALLERES / CENTROS DE SIMULACIÓN – PARA DOCENTES</b>	
<b>CARRERA:</b> COMPUTACIÓN/INGENIERÍA DE SISTEMAS		<b>ASIGNATURA:</b> PROGRAMACIÓN APLICADA	
<b>NRO. PROYECTO:</b>	1.1	<b>TÍTULO PROYECTO:</b> Practica de Arduino Desarrollo e implementación de un entorno de simulación Online para placas Arduino y electronica basica.	
<b>OBJETIVO:</b> Reforzar los conocimientos adquiridos en clase sobre Arduino.			
<b>INSTRUCCIONES:</b>		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 <a href="https://www.tinkercad.com">https://www.tinkercad.com</a> para simular circuitos electricos.	
		4. Revisar los siguientes videos que le ayudaran para realizar la tarea: - <a href="https://www.youtube.com/watch?v=r25dG32IWSU">https://www.youtube.com/watch?v=r25dG32IWSU</a> (Video de Electrónica Básica) - <a href="https://www.youtube.com/watch?v=hZmSG-IALAM">https://www.youtube.com/watch?v=hZmSG-IALAM</a> (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 <b>PDF</b> y los archivos al GitPersonal.  <b>Fecha de Entrega: 31 de Enero 2021</b>	
<b>ACTIVIDADES POR DESARROLLAR</b>			

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 auto fantástico que se prenda y se apague desde un pulsante.

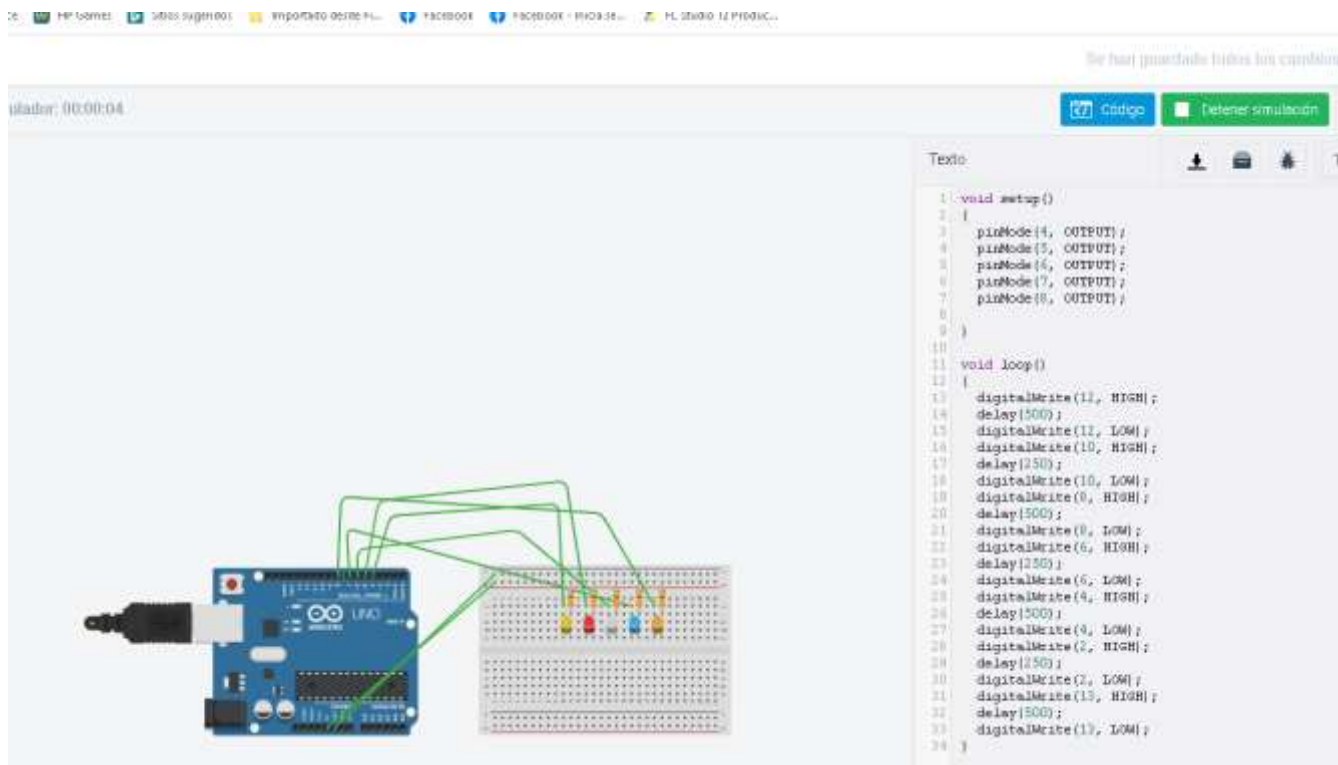
### Código:

```
void setup()
{
  pinMode(4, OUTPUT);
  pinMode(5, OUTPUT);
  pinMode(6, OUTPUT);
  pinMode(7, OUTPUT);
  pinMode(8, OUTPUT);
}

void loop()
{
  digitalWrite(12, HIGH);
  delay(500);
  digitalWrite(12, LOW);
  digitalWrite(10, HIGH);
  delay(250);
  digitalWrite(10, LOW);
  digitalWrite(8, HIGH);
  delay(500);
  digitalWrite(8, LOW);
  digitalWrite(6, HIGH);
  delay(250);
  digitalWrite(6, LOW);
  digitalWrite(4, HIGH);
  delay(500);
  digitalWrite(4, LOW);
  digitalWrite(2, HIGH);
  delay(250);
  digitalWrite(2, LOW);
  digitalWrite(13, HIGH);
  delay(500);
  digitalWrite(13, LOW);
}
```

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

## Circuito:



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.

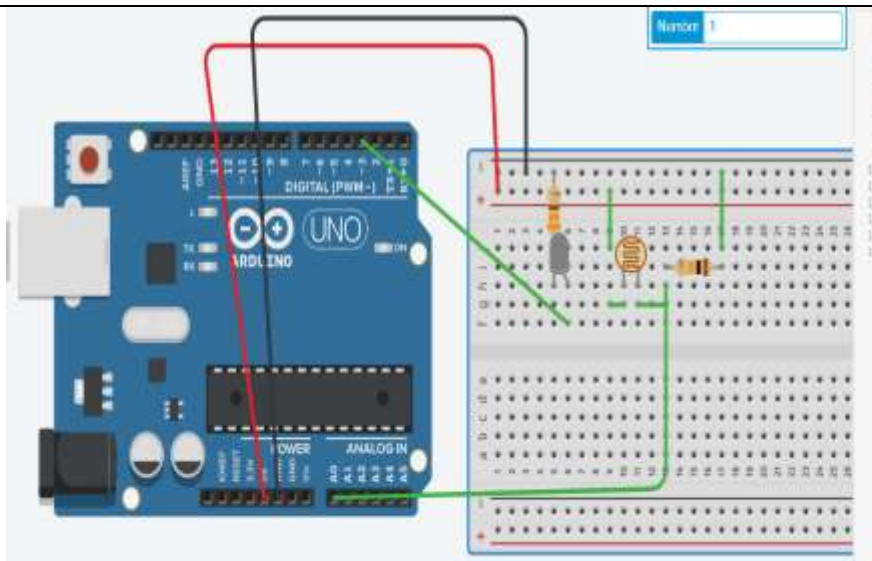
## Código:

```
#define LED 13
#define LDR A5

void setup()
{
  pinMode(LED, OUTPUT);
}

void loop()
{
  if (analogRead(LDR) < 900) digitalWrite(LED, HIGH);
  else digitalWrite(LED, LOW);
  delay(500);
}
```

## Circuito:



```

1 #define LED 1
2 #define LED 2
3
4 void setup()
5 {
6   pinMode(LED, OUTPUT);
7 }
8
9 void loop()
10 {
11   if (analogRead(A0) < 900) digitalWrite(LED, HIGH);
12   else digitalWrite(LED, LOW);
13   delay(1000);
14 }

```

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

#### Código:

```

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

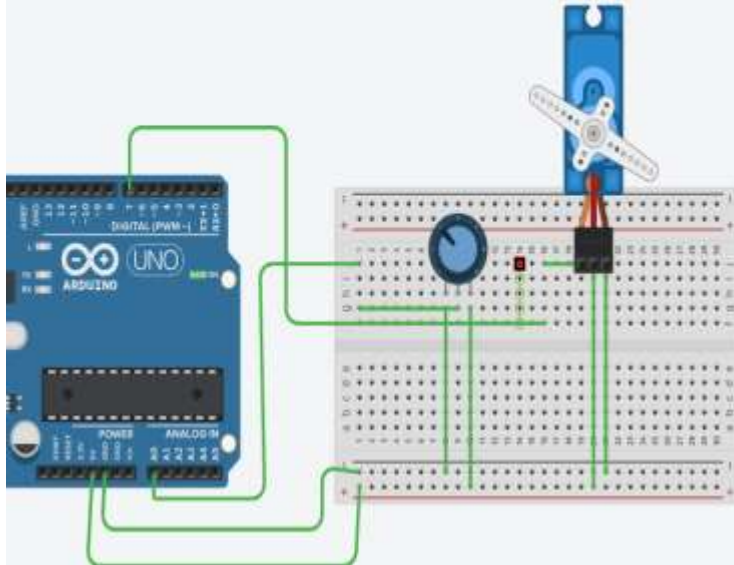
void setup()
{
  Serial.begin(9600);
  helice.attach(2);
}

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

```

#### Circuito:

en simulador: 00:00:42



Texto

```
1 #include <Servo.h>
2 Servo helice;
3 int grados;
4
5
6 void setup()
7 {
8   Serial.begin(9600);
9   helice.attach(7);
10 }
11
12 void loop()
13 {
14   grados = map(analogRead(A0), 0, 1023, 0, 180);
15   Serial.println(grados);
16   delay(200);
17   helice.write(grados);
18 }
```

### 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 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:** \_\_\_\_\_

**Nombre de estudiante:** JUAN JOSE CORDOVA CALLE

**Firma de estudiante:**

A handwritten signature in dark ink on a light background. The signature is stylized, starting with a long horizontal stroke that curves upwards and then forms several loops and curves before ending.