
	<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 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:** \_\_\_\_\_

 <b>UNIVERSIDAD POLITÉCNICA SALESIANA</b> ECUADOR	<b>Computación</b>	<b>Docente: Diego Quisi Peralta</b>
	Programacion Aplicada	<b>Período Lectivo:</b> Septiembre 2020 – Febrero 2021

CARRERA:

ASIGNATURA:

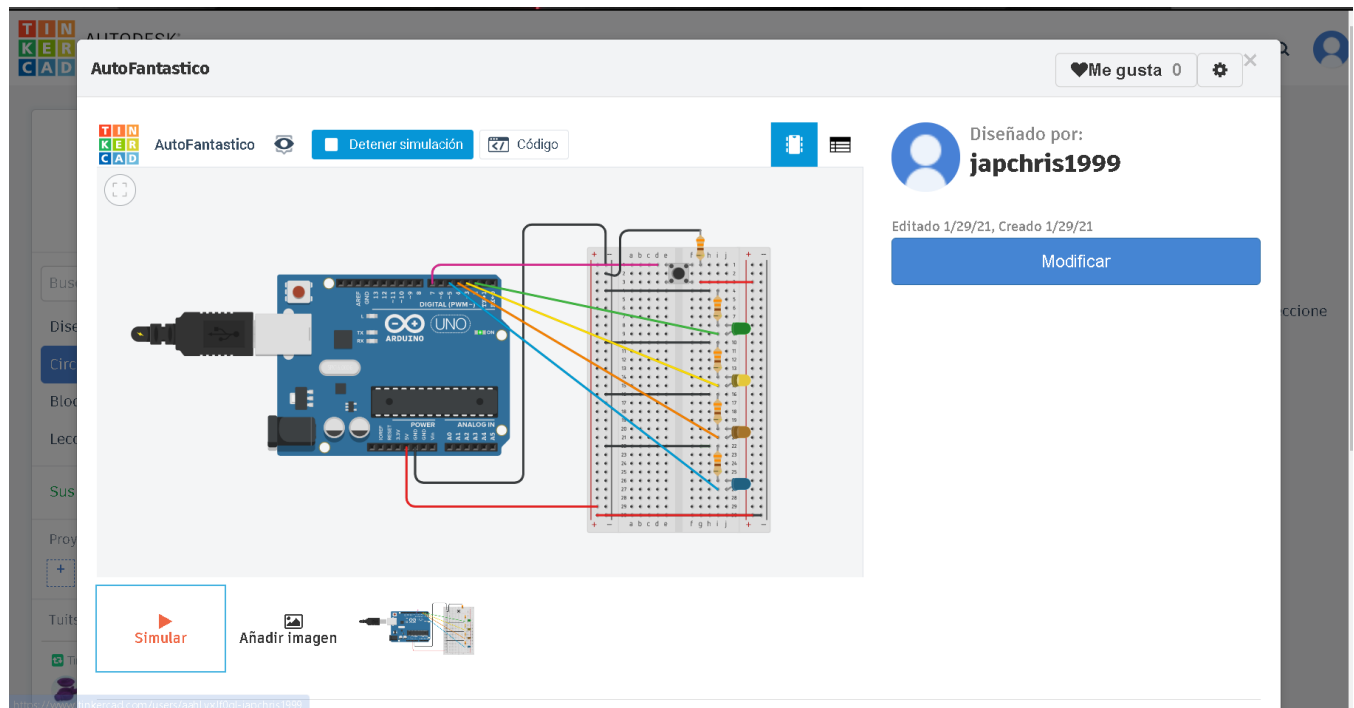
NRO. PRÁCTICA:

TÍTULO PRÁCTICA:

OBJETIVO ALCANZADO:

### ACTIVIDADES DESARROLLADAS

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



2.  
int est//1. Generar un autofantastico que se prenda  
//y se apague desde un pulsante.

```
int estado=0;
```

```
void setup(){
```

```
pinMode(2, OUTPUT);
```

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

```
pinMode(4, OUTPUT);
```

```
pinMode(5, OUTPUT);
```

```
pinMode(7,INPUT);

}

void loop(){

    estado =digitalRead(7);
    //Si el pulsante es soltado
    //volvera al estado inicial
    //pero terminara la secuencia
    if(estado == HIGH){
        secuencia1();
        secuencia2();
    }

}

//Secuencia para primer recorrido de encendido de luces
void secuencia1(){
    for(int i=2; i<6; i++){
        digitalWrite(i, HIGH);
        delay(400);
        digitalWrite(i, LOW);
    }
}

//Secuencia para el recorrido de regreso de luces

void secuencia2(){
    for(int i=5; i>1; i--){
        digitalWrite(i, HIGH);
        delay(400);
```

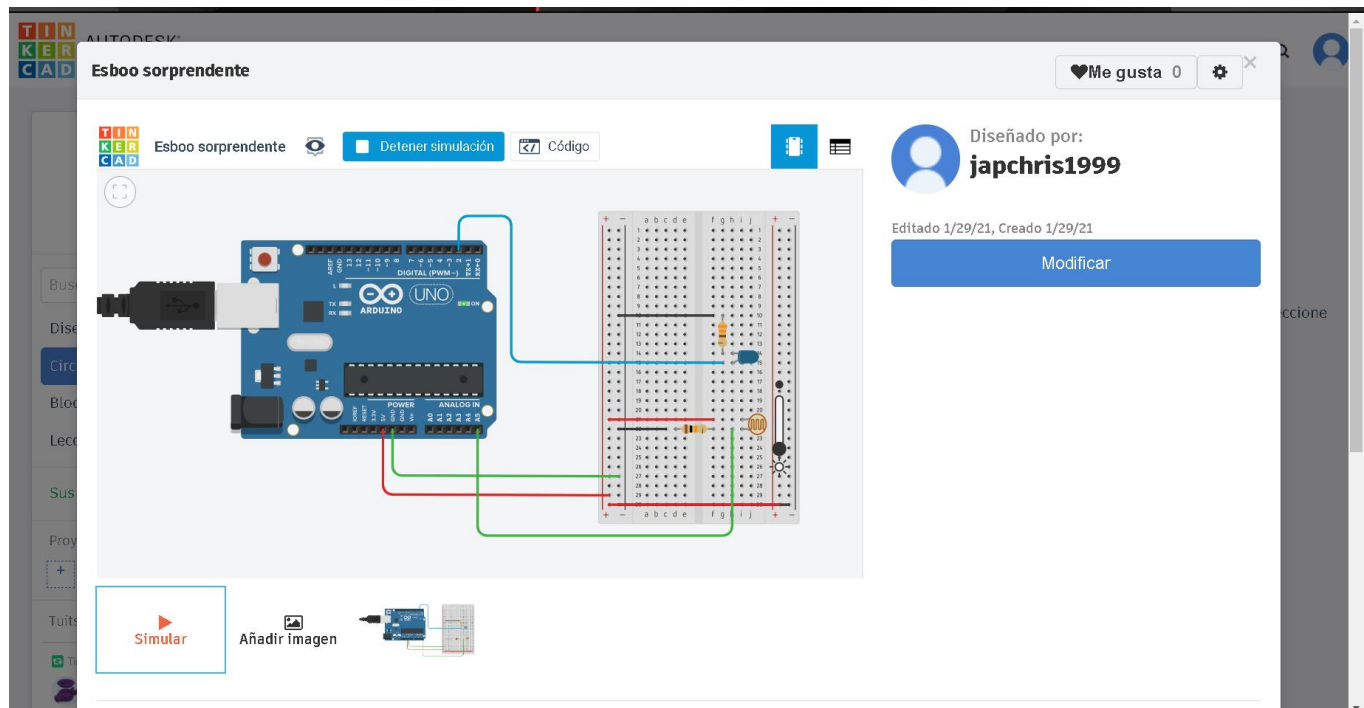
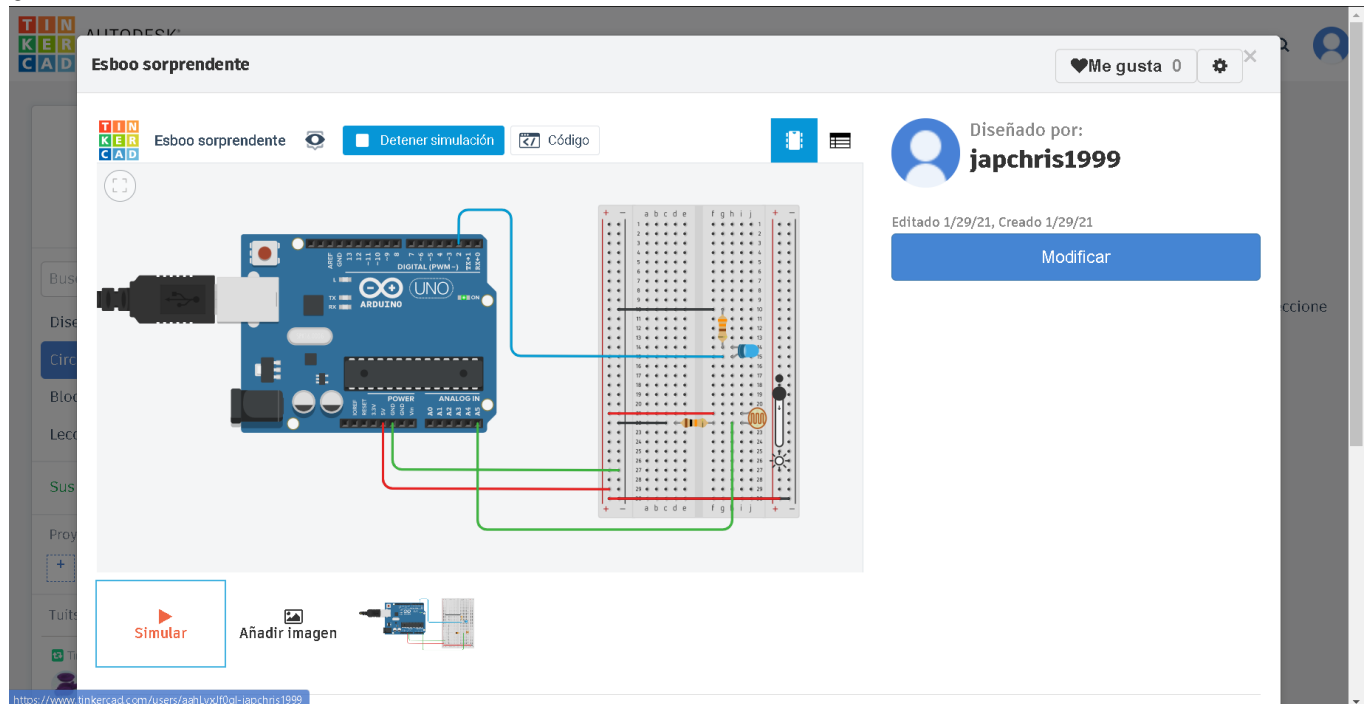
**digitalWrite(i, LOW);**

**}**

**}**

3. 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.

0



```
4.
int LED= 2;

int LDR= A5;

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

void loop(){

  if(analogRead(LDR) < 920){
    digitalWrite(LED, HIGH);
  }else{
    digitalWrite(LED, LOW);
  }
}
```

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

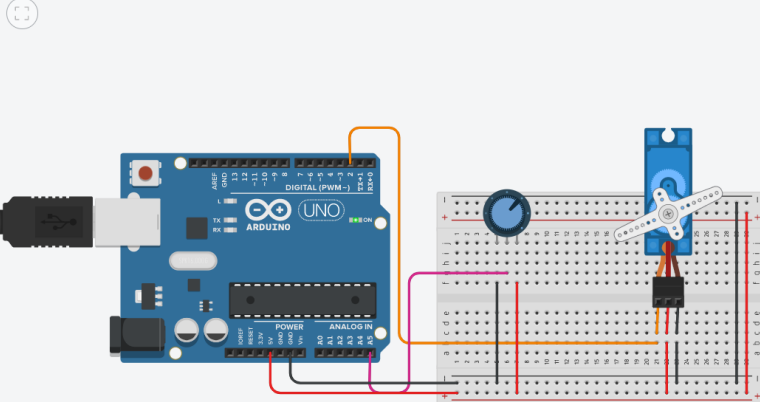
Sizzling Jofo-Bombul

Se han guardado todos los cambios.

Hora de simulador: 00:00:21

Código Detener simulación Exportar Compartir

1 (Arduino Uno R3)



```
1 #include <Servo.h>
2 Servo palanca;
3 int grados;
4 void setup(){
5   Serial.begin(9600);
6   palanca.attach(2);
7 }
8
9 void loop(){
10
11   grados = map (analogRead(A5),0,1023,0,180);
12   Serial.println(grados);
13   delay(250);
14   palanca.write(grados);
15
16 }
```

Monitor en serie

Env. Borrar

```
6. #include <Servo.h>

Servo palanca;

int grados;
```

```

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

void loop(){

  grados = map (analogRead(A5),0,1023,0,180);
  Serial.println(grados);
  delay(250);
  palanca.write(grados);

}

```

N.

#### RESULTADO(S) OBTENIDO(S):

El uso del simulador fue muy útil además de tener referencias en la web para su mejor uso, tanto a nivel de código como de manipulación de los componentes, los 3 enunciados que se propusieron fueron correctamente simulados

#### CONCLUSIONES:

El simulador es en toda regla lo mejor es como si lo tuviéramos en físico, su manejo es intuitivo.

#### RECOMENDACIONES:

Nombre de estudiante: \_\_\_\_\_ Christian Japon \_\_\_\_\_

Firma de estudiante: \_\_\_\_\_

