
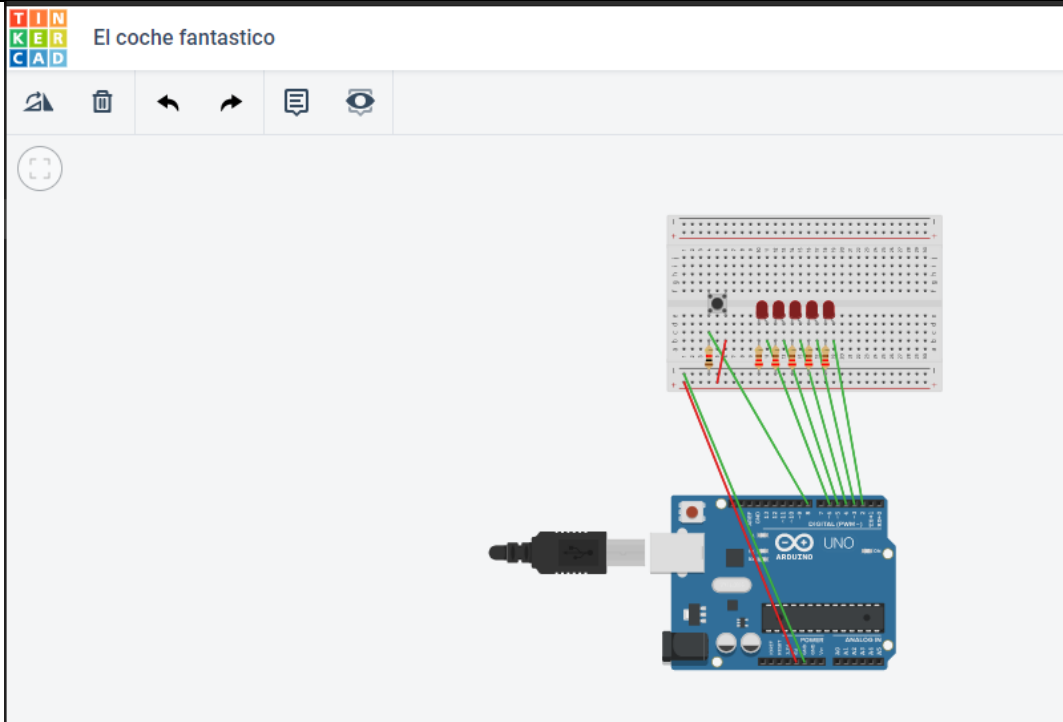
	<b>VICERRECTORADO DOCENTE</b>	<b>Código:</b> GUIA-PRL-001
	CONSEJO ACADÉMICO	<b>Aprobación:</b> 2016/04/06
<b>Formato:</b> Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

		<b>PRÁCTICA DE LABORATORIO</b>	
<b>CARRERA:</b> COMPUTACION		<b>ASIGNATURA:</b> PROGRAMACION APLICADA	
<b>NRO. PROYECTO:</b>	1.1	<b>TÍTULO PRÁCTICA:</b> Practica de Arduino  Desarrollo e implementación de un entorno de simulación Online para placas Arduino y electronica basica.	
<b>OBJETIVO ALCANZADO:</b>  Reforzar los conocimientos adquiridos en clase sobre Arduino.			
<b>ACTIVIDADES DESARROLLADAS</b>			
<p>1. Investigue, diseñe y desarrolle e implemente tres sistemas de simulación electrónica de Arduino dentro de la herramienta online Thincad.</p> <p>DEFINICIÓN DEL PROBLEMA:</p> <p><b>TinkerCad es un software gratuito</b> para el diseño 3D desarrollado por Autodesk. En su apartado circuits ofrece un <b>simulador online de Arduino</b> bastante completo y fácil de utilizar.</p> <p><b>TinkerCad ofrece bastantes componentes</b> 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.).</p> <p>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.</p> <p>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.</p> <p>En base a ello se propone resolver tres problemas electrónicos:</p> <p><b>1. Generar un auto fantástico que se prenda y se apague desde un pulsante.</b></p> <p>A continuación, se presenta el circuito armado en Tinkercad.</p>			



A continuación, se presenta el código:

```
short ledPins[] = {3, 4, 5, 6, 7};
short numberOfLeds = sizeof(ledPins) / sizeof(ledPins[0]);
short buttonPin = 2;
short lightingMode = 0;
bool movesToRight = true;
short currentLedIndex = 0;
bool previousButtonState = false;
bool currentButtonState = false;
long previousLightingStart;

void setup()
{
    Serial.begin(9600);
    // Setup pins.
    for (short i = 0; i < numberOfLeds; i++)
    {
        pinMode(ledPins[i], OUTPUT);
    }
    pinMode(buttonPin, INPUT);
    previousLightingStart = millis();
}
```

```
void loop()
{
    currentButtonState = digitalRead(buttonPin);
    // Switch modos.
    if (!previousButtonState
    && currentButtonState)
    {
        lightingMode++;
        if (lightingMode > 2)
        {
            lightingMode = 0;
        }
        previousLightingStart = millis();
    }

    // LED.
    switch (lightingMode)
    {
        // Slow.
        case 0:
            kitt(100);
            break;
        // Fast.
        case 1:
            kitt(60);
            break;
        // Off.
        case 2:
            for (short i = 0; i < numberOfLeds; i++)
            {
                digitalWrite(ledPins[i], LOW);
            }
            break;
    }

    previousButtonState = currentButtonState;
}

void kitt(short durationPerLed)
{
    if (millis() - previousLightingStart >= durationPerLed)
    {
        previousLightingStart = millis();
        if (movesToRight)
        {
            currentLedIndex++;
            if (currentLedIndex > numberOfLeds - 1)
            {
                currentLedIndex = numberOfLeds - 2;
            }
        }
    }
}
```

```
        movesToRight = false;
    }
}
else
{
    currentLedIndex--;
    if (currentLedIndex < 0)
    {
        currentLedIndex = 1;
        movesToRight = true;
    }
}
}

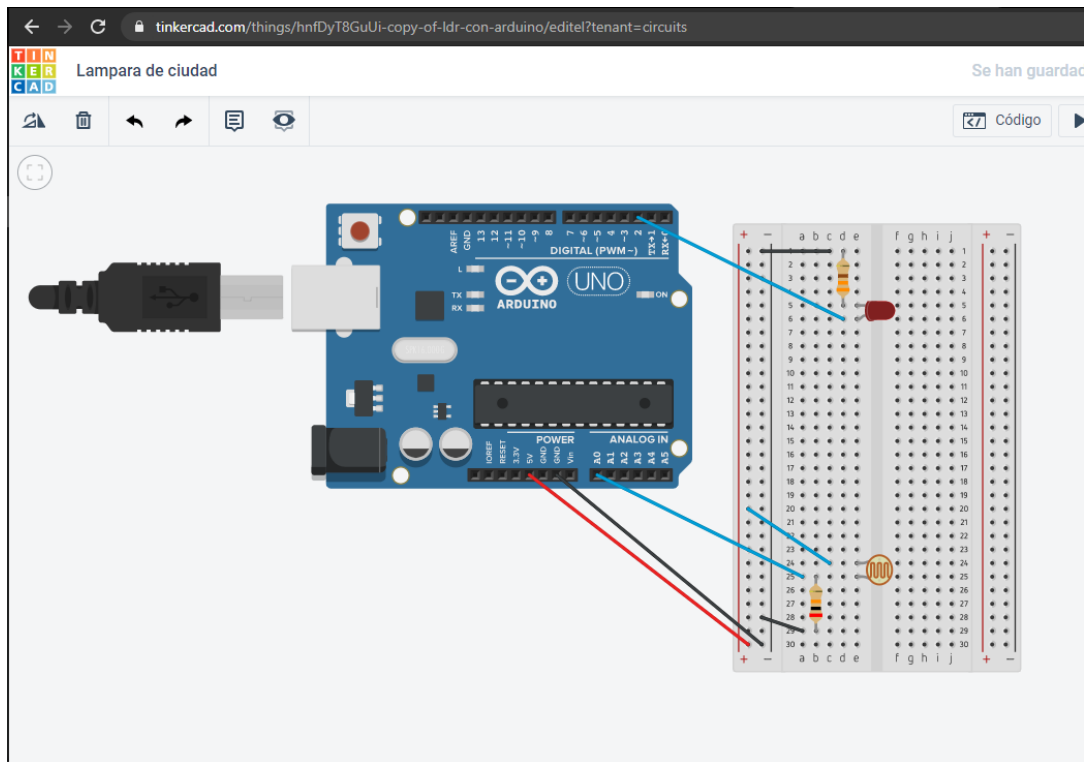
for (short i = 0; i < numberOfLeds; i++)
{
    if (i == currentLedIndex)
    {
        digitalWrite(ledPins[i], HIGH);
    }
    else
    {
        digitalWrite(ledPins[i], LOW);
    }
}
}
```

Finalmente, estos fueron los componentes utilizados

Nombre	Cantidad	Componente
U1	1	Arduino Uno R3
D1 D2 D3 D4 D5	5	Rojo LED
S1	1	Pulsador
R1 R6 R7 R8 R9	5	220 $\Omega$ Resistencia
R2	1	1 k $\Omega$ Resistencia


**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.**

A continuación, se presenta el circuito armado en Tinkercad.



A continuación, se presenta el código:

```
void setup()
```

	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

```

{
  pinMode(A0, INPUT);
  Serial.begin(9600);

  pinMode(2, OUTPUT);
}

void loop()
{
  while (1 == 1) {
    Serial.println(analogRead(A0));
    if (analogRead(A0) < 800) {
      digitalWrite(2, LOW);
    } else {
      digitalWrite(2, HIGH);
    }
    delay(1000);
  }
}


```

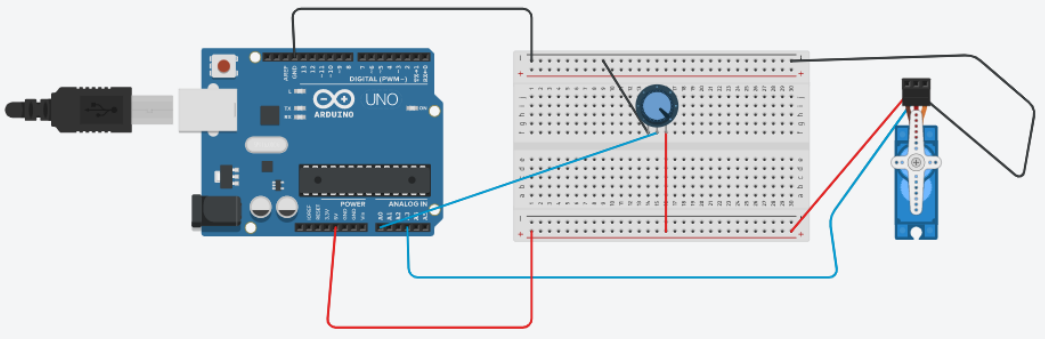
Finalmente, estos fueron los componentes utilizados

Cantidad	Componente
1	Arduino Uno R3
1	Rojo LED
1	330 $\Omega$ Resistencia
1	Fotorresistencia
1	20 k $\Omega$ Resistencia

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

A continuación, se presenta el circuito armado en Tinkercad.

	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		



A continuación, se presenta el código:

```

int pot = 0;


Servo servo_A3;

void setup()
{
  pinMode(A0, INPUT);
  servo_A3.attach(A3, 500, 2500);
}

void loop()
{
  pot = analogRead(A0);
  servo_A3.write(map(pot, 0, 1023, 0, 360));
  delay(10); // Delay a little bit to improve simulation performance
}

```

Finalmente, estos fueron los componentes utilizados

	<b>VICERRECTORADO DOCENTE</b>	<b>Código:</b> GUIA-PRL-001
	CONSEJO ACADÉMICO	<b>Aprobación:</b> 2016/04/06
<b>Formato:</b> Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

Cantidad	Componente
1	Arduino Uno R3
1	Microservomotor
1	250 k $\Omega$ Potenciómetro

**RESULTADO(S) OBTENIDO(S):**

- Familiarizarse con la herramienta de tinkercad.
- Entender cómo funciona la programación de Arduino.
- Desarrollar cosas practicas usando Arduino.
- Investigar sobre comando de Arduino.

**CONCLUSIONES:**

En conclusión, esta practica ha sido de gran ayuda para entender cómo funciona Arduino sin tener la necesidad de comprar componente y hacerlo solamente en línea. Además, es de gran ayuda para adquirir conocimientos sobre esta unidad en la materia de Programación aplicada.

**RECOMENDACIONES:**

No existe ninguna recomendación de mi parte.

**Nombre de estudiante:** Adolfo Sebastián Jara Gavilanes.

**Firma de estudiante:**

