



Inteligencia Artificial Avanzada para Ciencia de Datos

Instituto Tecnológico de Estudios Superiores de Monterrey

Documento Evidencias Semanales

Rodrigo Montelongo, A00827757

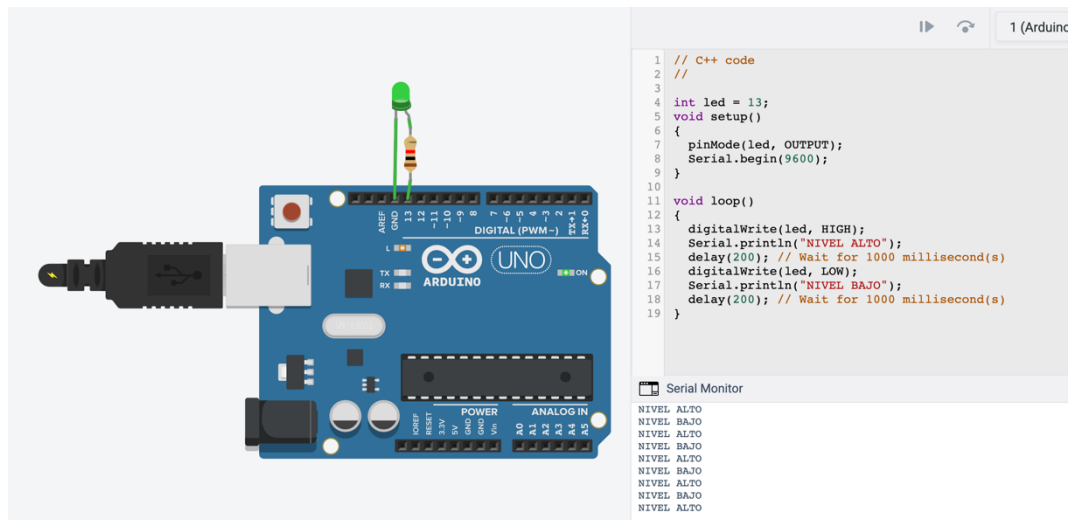
7 de septiembre del 2022

Link al video:

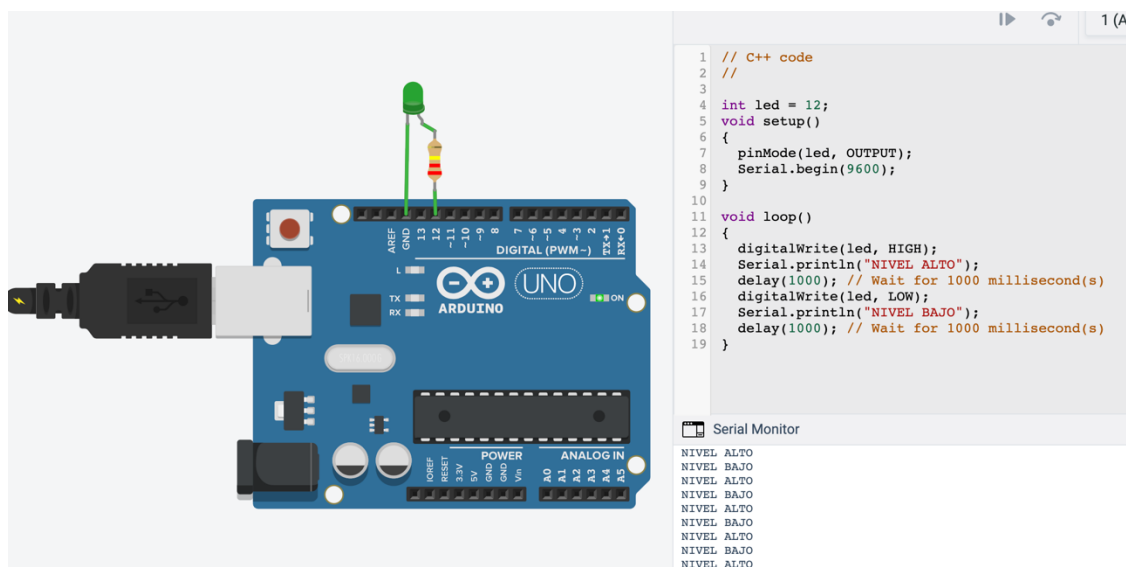
<https://drive.google.com/file/d/1nXeml296302CoyY5ctEEuFex7-eDey5y/view?usp=sharing>

Semana 1

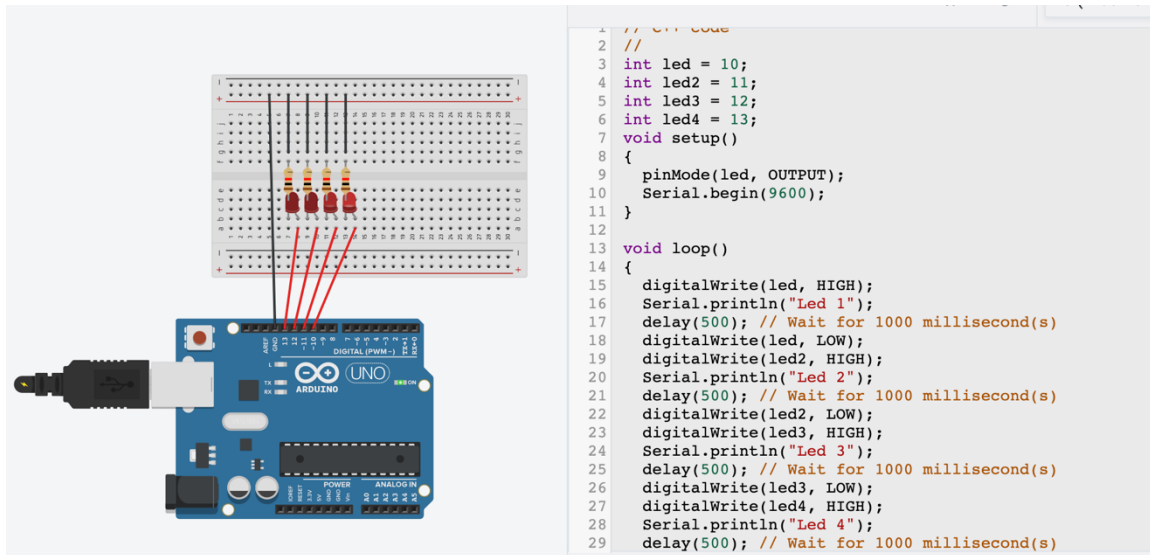
1. Simular el esquema propuesto.



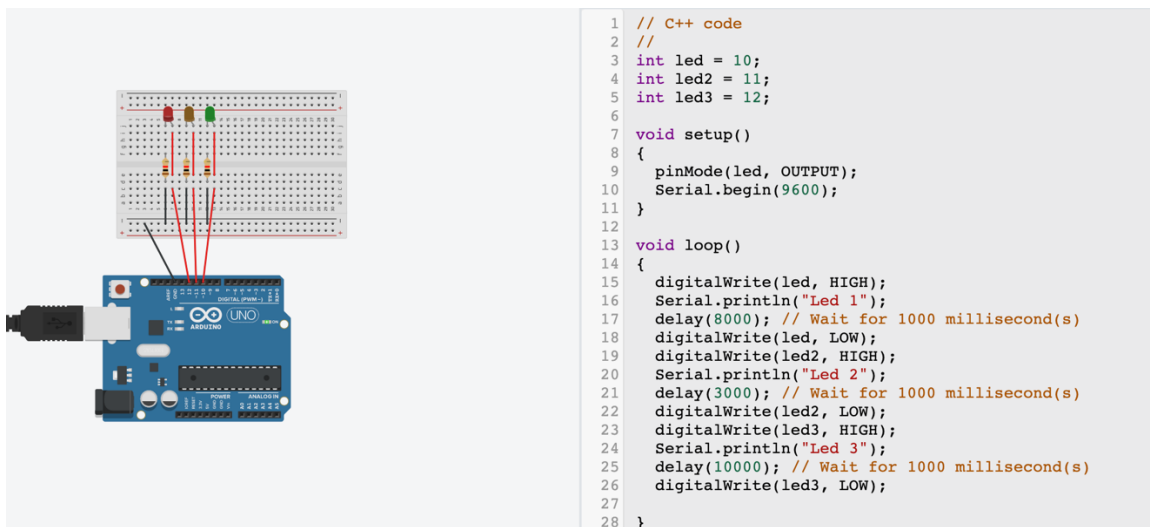
2. Modificar el circuito para para que el led se encienda al conectarlo al pin 12 y modificar el periodo de parpadeo de 200 ms a 1 s.



3. Realizar un programa que ejecute el encendido correlativo de cuatro LEDs conectados a los pins 10,11,12 y 13 con un periodo de 500 ms.



4. Realizar el programa que simule el comportamiento de un semáforo. Emplear para ello tres LEDs de colores rojo, verde y naranja. La secuencia que deben seguir es la siguiente: Verde (8 segundos) – Naranja (3 segundos) – Rojo (10 segundos).

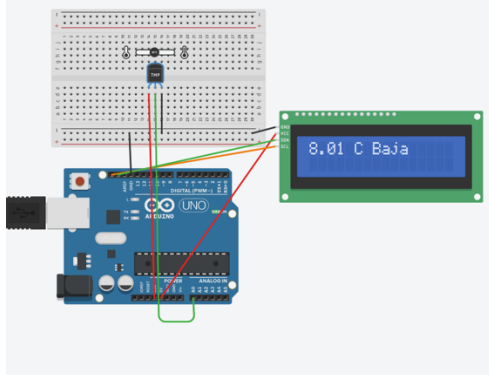


Semana 2

Crear una propuesta de forma integrada para utilizar por lo menos 3 soluciones de las actividades presentadas en la semana 2 (1 sensor, 1 microcontrolador, 1 pantalla LCD), crear una condición para verificar y presentar mensajes en la pantalla LCD.

Temperature Sensor [TMP36]

Name 3



```
1 #include <Adafruit_LiquidCrystal.h>
2
3 #define sensorPin A0
4 // C++ code
5 //
6 Adafruit_LiquidCrystal lcd_1(0);
7 void setup()
8 {
9   lcd_1.begin(16,2);
10   Serial.begin(9600);
11 }
12 void loop()
13 {
14   int inputTemperatura = analogRead(sensorPin);
15   float voltaje = inputTemperatura * (3.3 / 1024.0);
16   float temperaturaCelsius = (voltaje - 0.5) * 100;
17   lcd_1.print(temperaturaCelsius);
18   lcd_1.print(" C");
19   if (temperaturaCelsius <= 15){
20     lcd_1.print(" Baja");
21   } else if (temperaturaCelsius > 15 & temperaturaCelsius <= 30){
22     lcd_1.print(" Normal");
23   } else if (temperaturaCelsius > 30){
24     lcd_1.print(" Alta");
25   }
26   delay(2000);
27   lcd_1.clear();
28 }
```

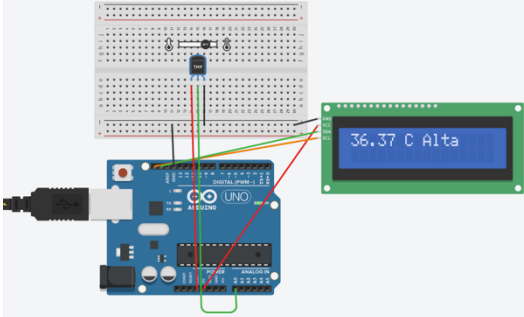
Temperature Sensor [TMP36]

Name 3



```
1 #include <Adafruit_LiquidCrystal.h>
2
3 #define sensorPin A0
4 // C++ code
5 //
6 Adafruit_LiquidCrystal lcd_1(0);
7 void setup()
8 {
9   lcd_1.begin(16,2);
10   Serial.begin(9600);
11 }
12 void loop()
13 {
14   int inputTemperatura = analogRead(sensorPin);
15   float voltaje = inputTemperatura * (3.3 / 1024.0);
16   float temperaturaCelsius = (voltaje - 0.5) * 100;
17   lcd_1.print(temperaturaCelsius);
18   lcd_1.print(" C");
19   if (temperaturaCelsius <= 15){
20     lcd_1.print(" Baja");
21   } else if (temperaturaCelsius > 15 & temperaturaCelsius <= 30){
22     lcd_1.print(" Normal");
23   } else if (temperaturaCelsius > 30){
24     lcd_1.print(" Alta");
25   }
26   delay(2000);
27   lcd_1.clear();
28 }
```

Name 3



```

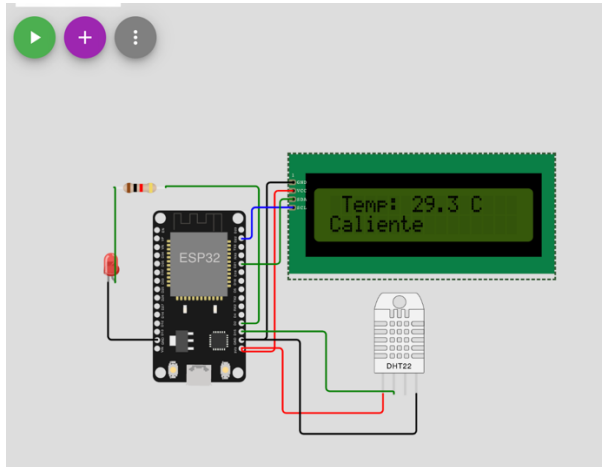
1 #include <Adafruit_LiquidCrystal.h>
2
3 #define sensorPin A0
4 // C++ code
5 //
6 Adafruit_LiquidCrystal lcd_1(0);
7 void setup()
8 {
9   lcd_1.begin(16,2);
10  Serial.begin(9600);
11 }
12 void loop()
13 {
14   int inputTemperatura = analogRead(sensorPin);
15   float voltaje = inputTemperatura * (3.3 / 1024.0);
16   float temperaturaCelsius = (voltaje - 0.5) * 100;
17   lcd_1.print(temperaturaCelsius);
18   lcd_1.print(" C");
19   if (temperaturaCelsius <= 15){
20     lcd_1.print(" Baja");
21   } else if (temperaturaCelsius > 15 & temperaturaCelsius <= 30)
22     lcd_1.print(" Normal");
23   } else if (temperaturaCelsius > 30){
24     lcd_1.print(" Alta");
25   }
26   delay(2000);
27   lcd_1.clear();
28 }

```

Semana 3

Especificaciones:

- 1) Presentar una solución que utiliza:
 - a. Un microcontrolador
 - b. 1 led
 - c. 1 pantalla LCD
 - d. Un sensor por elegir
- Identificar un código para el sensor (número ficticio)
 - Presentar la lectura de los datos del sensor en la pantalla y enviar para la base de Firebase, con la fecha, código del sensor, valor del sensor, información sobre el estatus.
 - Crear una condición para validar los valores
 - LED parpadea por 5 segundos para informar que se tienen datos en transmisión.



<https://miproyectoiot-565f5-default-rtdb.firebaseio.com>

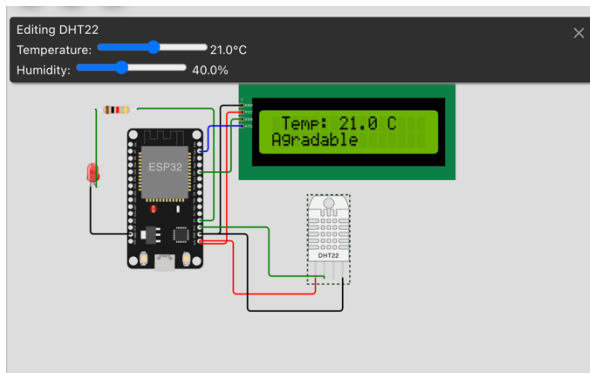
<https://miproyectoiot-565f5-default-rtdb.firebaseio.com/>

codigo: xyz ABC X

estatus: "Caliente"

fecha: "12-09-2022"

valor: "29.3"



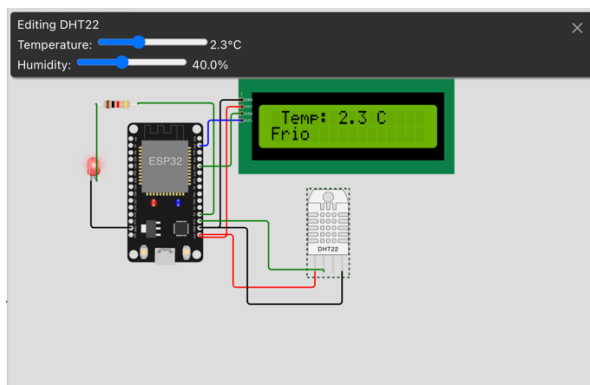
<https://miproyectoiot-565f5-default-rtdb.firebaseio.com/>

codigo: xyz ABC X

estatus: "Agradable"

fecha: "12-09-2022"

valor: "21.0"



<https://miproyectoiot-565f5-default-rtdb.firebaseio.com/>

codigo: xyz ABC X

estatus: "Frio"

fecha: "12-09-2022"

valor: "2.3"

main.py ●

diagram.json

ssd1306.py

i2c_lcd.py

lcd_api.py ▼

```
1
2 import machine
3 import json
4 import network
5 import urequests
6 from machine import I2C, Pin
7 from i2c_lcd import I2cLcd
8 import time
9 import dht
10
11 sensor = dht.DHT22(Pin(15))
12 wlan = network.WLAN(network.STA_IF)
13 wlan.active(True)
14 wlan.connect("Wokwi-GUEST", "")
15
16 while not wlan.isconnected():
17     pass
18
19
```

```
19 while True:
20
21     sensor.measure()
22     temp = sensor.temperature()
23
24     AddressOfLcd = 0x27
25     i2c = I2C(scl = Pin(22), sda = Pin(21), freq = 400000)
26     lcd = I2cLcd(i2c, AddressOfLcd, 2, 16)
27     led = machine.Pin(2, machine.Pin.OUT)
28
29     lcd.move_to(1,0)
30     lcd.putstr("Temp: " + str(temp) + " C")
31     lcd.move_to(0,1)
32     if temp <= 15:
33         estatus = "Frio"
34         lcd.putstr(estatus)
35     elif temp > 15 and temp <= 25:
36         estatus = "Agradable"
37         lcd.putstr(estatus)
38     elif temp > 25:
39         estatus = "Caliente"
40         lcd.putstr(estatus)
41
42     url = "https://miproyectoiot-565f5-default-rtdb.firebaseio.com/.json"
43     data = {"valor": str(temp), 'estatus' : estatus, 'fecha': '12-09-2022', 'codigo': 'xyz'}
44     r = urequests.patch(url, data = json.dumps(data))
45
46     led.value(1)
47     time.sleep(1)
48     led.value(0)
49     time.sleep(1)
50     time.sleep(5)
51
```

Semana 4 y 5

Especificaciones:

- Presentar una propuesta para la utilización de servicios de cloud para envío y recibimientos de datos con Firebase.
- Integrar la solución de la semana 3, utilizando los mismos componentes y reglas para validación de datos.
- Presentar en cloud lo envío y recibimientos de datos, también una presentación de la lectura de datos con gráficas y control de operaciones en su microcontrolador por medio de su solución Cloud, como por ejemplo: prender un Led, activar un motor o sensor via Cloud.

Act 4

Setup

Sketch

Metadata

Variables

ADD

	Name ↓	Last Value	Last Update	
<input type="checkbox"/>	estatus <code>String</code> estatus;	-		⋮
<input type="checkbox"/>	led_switch <code>bool</code> led_switch;	false	12 Sep 2022 20:29:11	⋮
<input type="checkbox"/>	temperature <code>float</code> temperature;	-		⋮

Associated Device

ESP_Rodrigo

ID: 20284c5f-5dd7-44bb-89b2-...

Type: SparkFun ESP32 Thing

Status: Offline

Change

Detach

Network

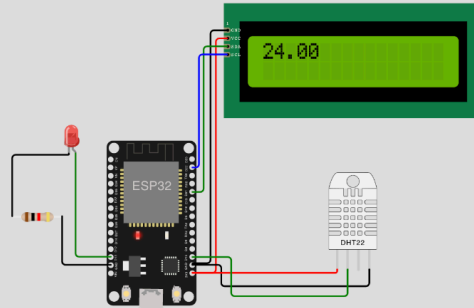
Wi-Fi Name: Wokwi-...

Password:

Secret Key:

Simulation

00:32.932 70%



***** Arduino IoT Cloud - configuration info *****

Device ID: 20284c5f-5dd7-44bb-89b2-1efb1ff035a7

MQTT Broker: mqttts-up.iot.arduino.cc:8884

WiFi status ESP: 255

Connected to "Wokwi-GUEST"

Connected to Arduino IoT Cloud

Thing ID: ele9fc01-5f1a-4602-b072-cacfad332e9a

estatus



Mon 12 Sep

Agradable 23:19

Type a message



temperature

24.000

led_switch

