



Tecnológico de Monterrey

TC3006C – Inteligencia artificial avanzada para la ciencia de datos I (Gpo 102)

Módulo 4 – Integración de hardware para la ciencia de datos

Documento final de evidencias de actividades del reto

Profesor: PhD. Antonio Carlos Bento

Edgar Castillo Ramírez

A00827826

Monterrey, Nuevo León, 12 de septiembre de 2022.

Índice

Semana 1	3
Especificaciones	3
Restricciones	3
Investigar	3
Experimentar	3
Interpretar	3
Aplicar	4
Auto-evaluación	7
Informar las referencias bibliográficas	8
Semana 2	8
Especificaciones	8
Restricciones	8
Investigar	8
Experimentar	8
Interpretar	9
Aplicar	11
Auto-evaluación	13
Informar las referencias bibliográficas	13
Semana 3	13
Especificaciones	13
Restricciones	14
Investigar	14
Experimentar	14
Interpretar	14
Aplicar	16
Auto-evaluación	18
Informar las referencias bibliográficas	19
Semana 4	19
Especificaciones	19

Restricciones	19
Investigar	19
Experimentar	19
Interpretar	20
Aplicar	22
Auto-evaluación	27
Informar las referencias bibliográficas	27
Semana 5	28

Semana 1

Especificaciones

Para esta semana, las especificaciones dadas por el profesor son las siguientes:

1. Simular el esquema propuesto con la herramienta online.
2. Presentar un circuito para que el LED se encienda al conectarlo al pin 12 y que el período de parpadeo sea de 1s (1000ms).
3. Realizar un programa que ejecute el encendido correlativo de cuatro LEDs conectados a los pines 10, 11, 12 y 13 con un período de 500ms.
4. Realizar un 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(8s), Naranja(3s), Rojo(10s).
5. De manera opcional, hacer parpadear el LED durante el estado amarillo con un período de 500ms.

Restricciones

Para esta semana, las restricciones presentadas por el profesor son las siguientes:

1. Utilizar C++
2. Utilizar Arduino Uno

Investigar

En la actividad de esta semana, no fue necesaria la investigación de contenido. El material visto en clase fue suficiente para generar una solución a las especificaciones.

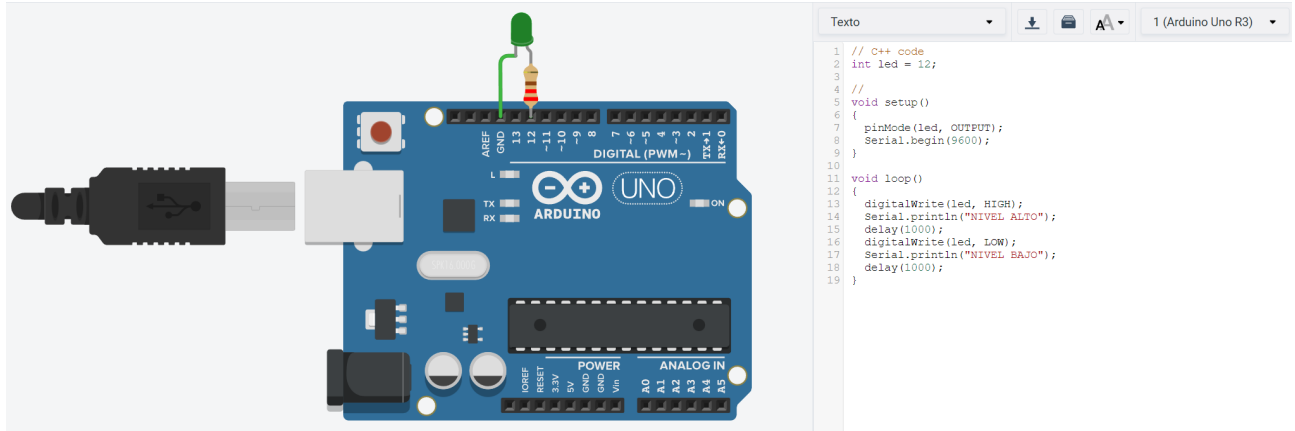
Experimentar

En esta semana, se pudo experimentar dentro de la clase realizando ejercicios de práctica, además de la explicación del profesor. Entre los ejercicios realizados estuvo:

- Introducción a Tinkercad Circuits Simulator
- Creación de un circuito sencillo de parpadeo de un LED.

Interpretar

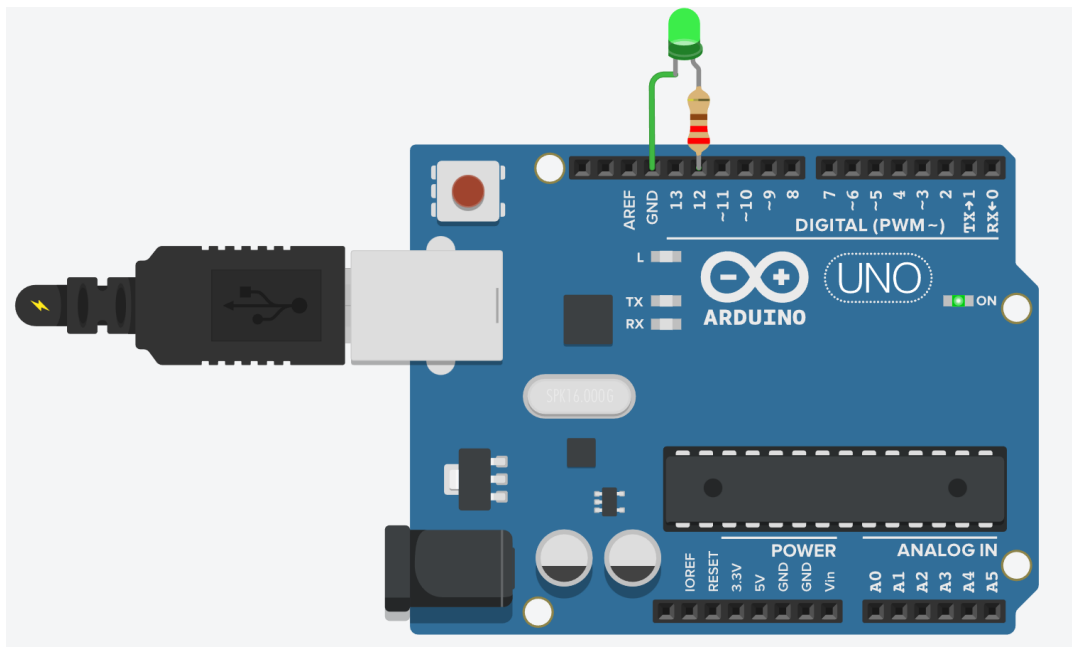
Después de generar un circuito sencillo, pude apreciar el funcionamiento de la plataforma de Tinkercad. De igual manera, comprendí que el microcontrolador se puede manejar por medio de código, por lo que gracias a esto pude generar mi primer circuito funcional en el simulador. Se observa la imagen a continuación:



Aplicar

Estas son las soluciones generadas para esta semana:


1. Presentar un circuito para que el LED se encienda al conectarlo al pin 12 y que el período de parpadeo sea de 1s (1000ms).



```

1  // C++ code
2  int led = 12;
3
4  //
5  void setup()
6  {
7      pinMode(led, OUTPUT);
8      Serial.begin(9600);
9  }
10
11 void loop()
12 {
13     digitalWrite(led, HIGH);
14     Serial.println("NIVEL ALTO");
15     delay(1000);
16     digitalWrite(led, LOW);
17     Serial.println("NIVEL BAJO");
18     delay(1000);
19 }

```

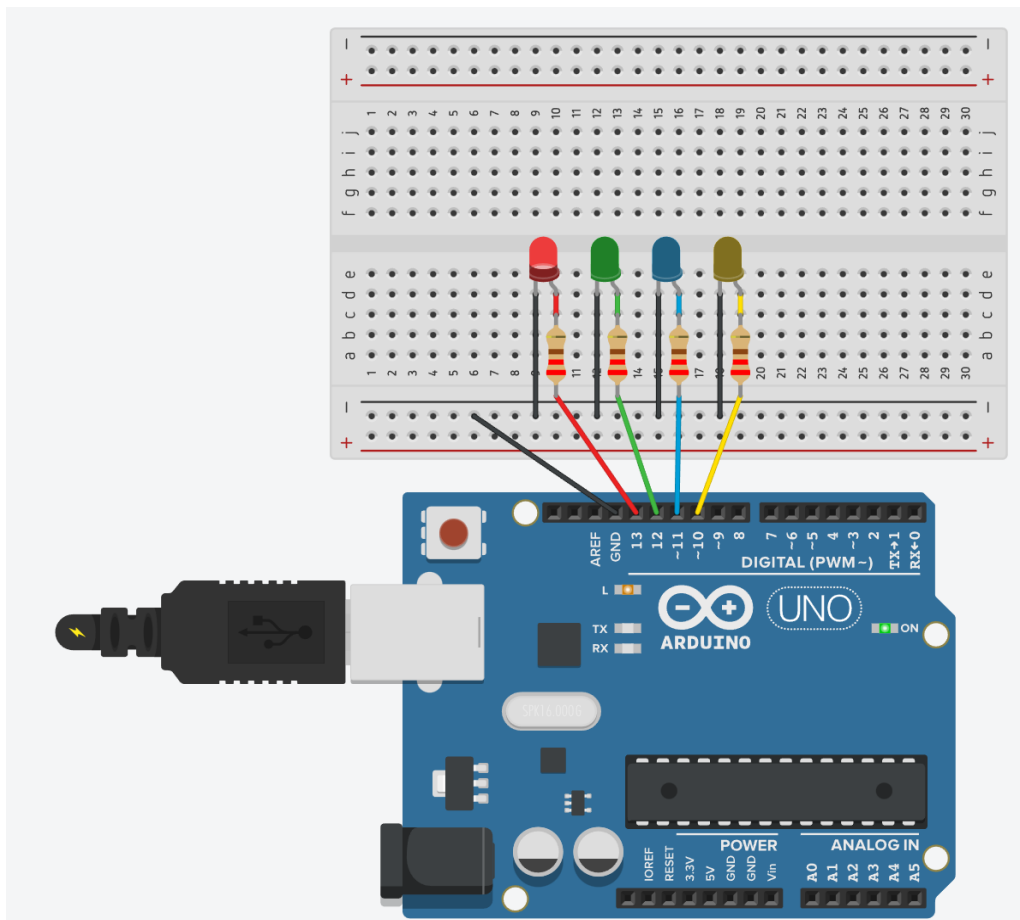
 Monitor en serie

```

NIVEL ALTO
NIVEL BAJO
NIVEL ALTO
NIVEL BAJO
NIVEL ALTO
NIVEL BAJO
NIVEL ALTO
NIVEL BAJO

```

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



```

1 // C++ code
2 int leds[4]={13,12,11,10};
3 String colors[4]={"Red", "Green", "Blue", "Yellow"};
4
5 //
6 void setup()
7 {
8     int i = 0;
9     for(i=0;i<4;i++){
10         pinMode(leds[i],OUTPUT);
11     }
12     Serial.begin(9600);
13 }
14
15 void loop()
16 {
17     int i = 0;
18     for(i=0;i<4;i++){
19         digitalWrite(leds[i], HIGH);
20         Serial.println(colors[i]);
21         delay(1000);
22         digitalWrite(leds[i], LOW);
23     }
24 }

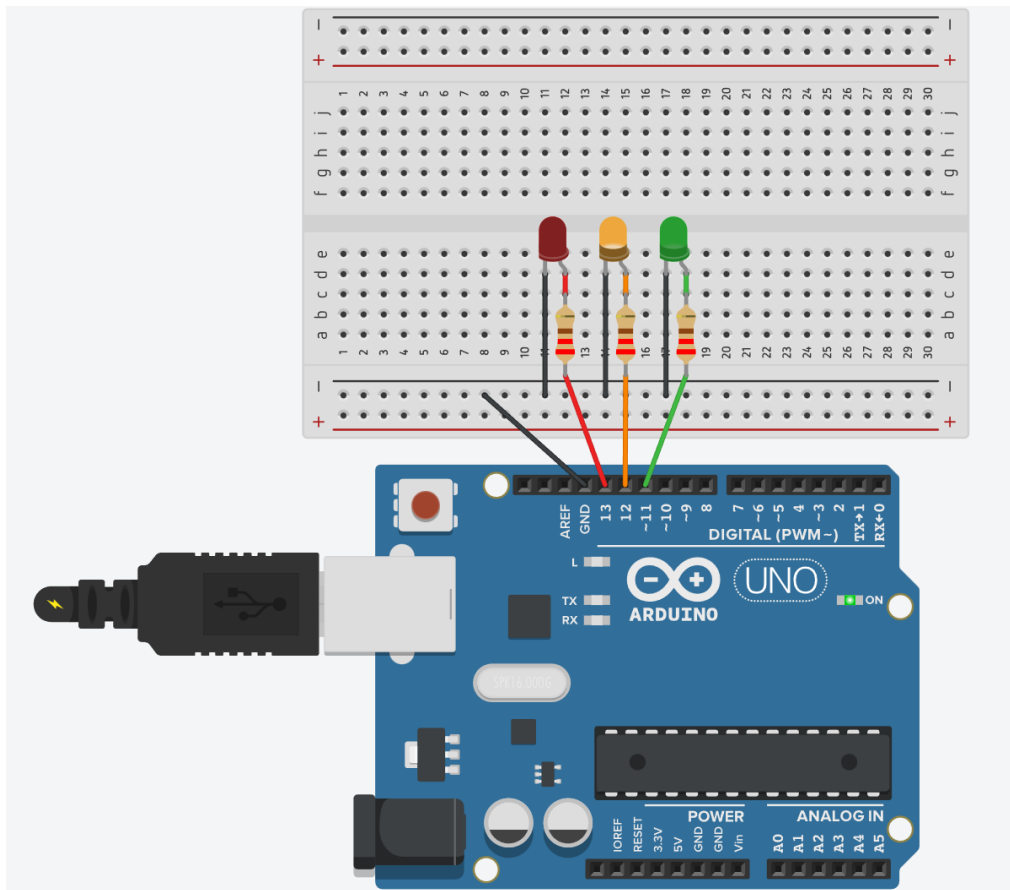
```



Monitor en serie

Green
Blue
Yellow
Red
Green
Blue
Yellow
Red


- Realizar un 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(8s), Naranja(3s), Rojo(10s).



```

1 // C++ code
2 //
3 int red = 13;
4 int orange = 12;
5 int green = 11;
6
7 void setup()
8 {
9     pinMode(red, OUTPUT);
10    pinMode(orange, OUTPUT);
11    pinMode(green, OUTPUT);
12    Serial.begin(9600);
13 }
14
15 void loop()
16 {
17     digitalWrite(green, HIGH);
18     Serial.println("GO");
19     delay(8000);
20     digitalWrite(green, LOW);
21
22     int i = 0;
23     for (i = 0; i < 3; i++){
24         digitalWrite(orange, HIGH);
25         Serial.println("CAUTION");
26         delay(500);
27         digitalWrite(orange, LOW);
28         delay(500);
29     }
30
31     digitalWrite(red, HIGH);
32     Serial.println("STOP");
33     delay(10000);
34     digitalWrite(red, LOW);
35
36 }

```

 Monitor en serie

CAUTION
CAUTION
STOP
GO
CAUTION
CAUTION
CAUTION
STOP

Auto-evaluación

Los resultados fueron positivos y correctos ya que se pudieron generar soluciones que satisfacen las especificaciones de esta semana.

Esto ayudará para la elaboración de soluciones en semanas próximas y poder cumplir con el indicador de la rúbrica.

Informar las referencias bibliográficas

Plataforma utilizada para la solución: <https://www.tinkercad.com/>

Semana 2

Especificaciones

Para esta semana, las especificaciones dadas por el profesor son las siguientes:

1. 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, por ejemplo:
 - Entre temperaturas de 0 y 25 grados, imprimir Temperatura Baja.
 - Entre temperaturas de 25 y 40 grados, imprimir Temperatura Agradable.
 - Más de 40 grados de temperatura, imprimir Temperatura Alta.

Restricciones

Para esta semana, las restricciones presentadas por el profesor son las siguientes:

1. Utilizar C++
2. Utilizar Arduino Uno
3. Utilizar solamente los componentes presentados durante la semana 2

Investigar

En la actividad de esta semana, no fue necesaria la investigación de contenido. El material visto en clase fue suficiente para generar una solución a las especificaciones.

Experimentar

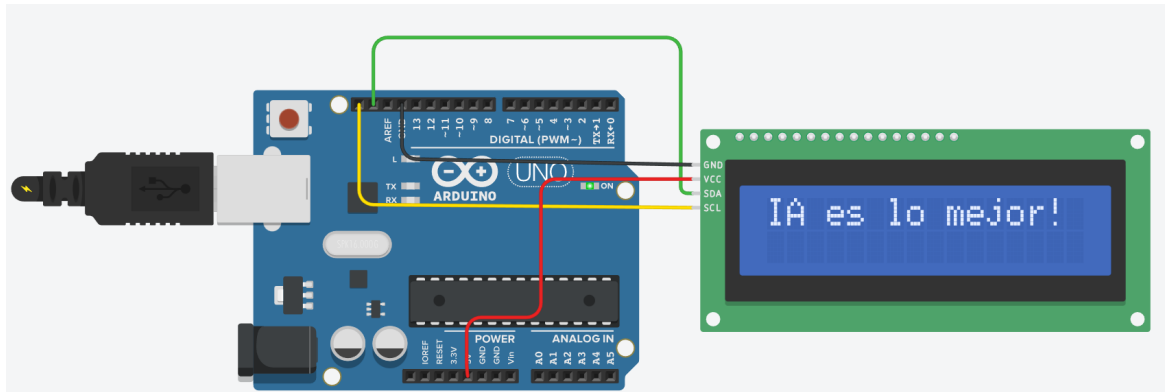
En esta semana, se pudo experimentar dentro de la clase realizando ejercicios de práctica, además de la explicación del profesor. Entre los ejercicios realizados estuvo:

- Creación de un circuito con pantalla LCD
- Creación de un circuito con sensor PIR
- Creación de un circuito con sensor IR
- Creación de un circuito con sensor de temperatura
- Creación de un circuito con sensor de Proximidad, distancia
- Creación de un circuito con sensor de Posición y velocidad

Interpretar

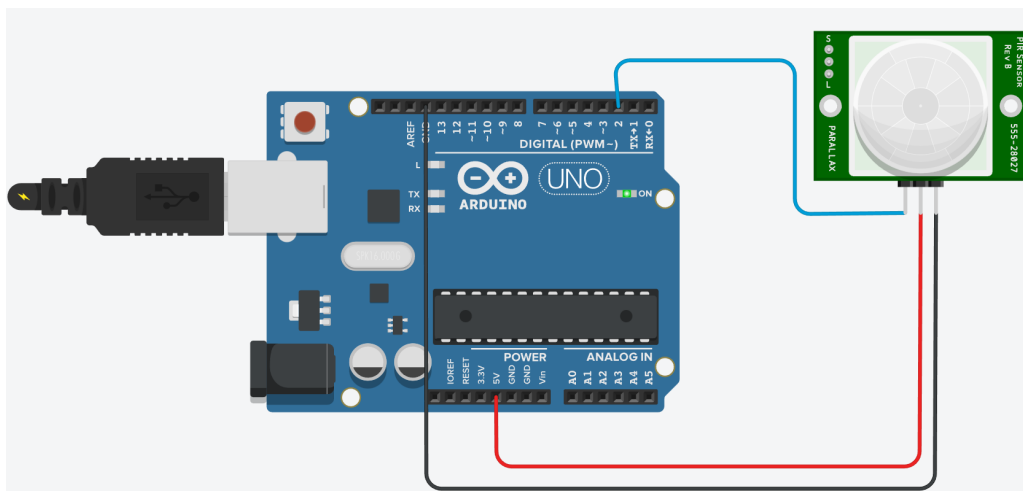
En esta semana se probaron algunos sensores y componentes en la clase para poder observar su funcionamiento en el simulador Tinkercad.

1. Creación de un circuito con pantalla LCD



```
1 // C++ code
2 #include <Adafruit_LiquidCrystal.h>
3 //
4
5 Adafruit_LiquidCrystal lcd_1(0);
6
7 void setup()
8 {
9   lcd_1.begin(16,2);
10 }
11
12 void loop()
13 {
14   lcd_1.print("Hola Mundo!!!");
15   delay(1000);
16   lcd_1.clear();
17   lcd_1.print("IA es lo mejor!");
18   delay(1000);
19   lcd_1.clear();
20 }
```


2. Creación de un circuito con sensor PIR



```

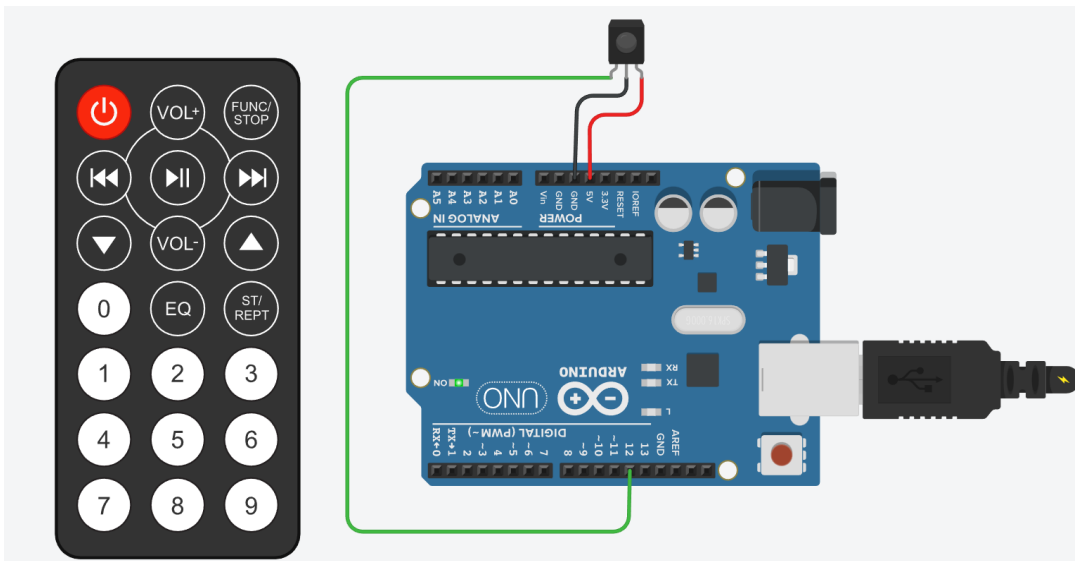
1 // C++ code
2 //
3 void setup()
4 {
5   Serial.begin(9600);
6   pinMode(2, INPUT);
7 }
8
9 void loop()
10 {
11   Serial.println(digitalRead(2));
12   delay(100);
13 }

```

 Monitor en serie

1
1
1
1
0
0
0
0

3. Creación de un circuito con sensor IR



```

1 #include <IRremote.h>
2 int IRpin = 12;
3 IRrecv irrecv(IRpin);
4 decode_results reading;
5
6 // C++ code
7 //
8 void setup()
9 {
10   Serial.begin(9600);
11   irrecv.enableIRIn();
12 }
13
14 void loop()
15 {
16   if(irrecv.decode(&reading)){
17     Serial.print("NEC: ");
18     Serial.print(reading.value, HEX);
19     Serial.print("\tBits: ");
20     Serial.println(reading.bits);
21     delay(100);
22     irrecv.resume();
23   }
24 }

```



Monitor en serie

The function decode(&results)) is de

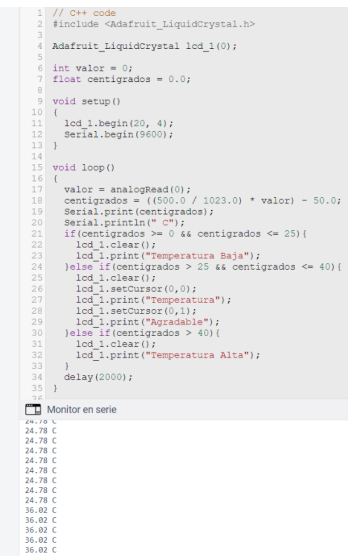
NEC: FD8877	Bits: 32
NEC: FD08F7	Bits: 32
NEC: FD08F7	Bits: 32
NEC: FD00FF	Bits: 32
NEC: FD00FF	Bits: 32
NEC: FFFFFFFF	Bits: 0
NEC: FFFFFFFF	Bits: 0
NEC: FD8877	Bits: 32
NEC: FD8877	Bits: 32

Por cuestiones de tiempo no me fue posible realizar la práctica de los sensores de temperatura, proximidad/distancia y posición/velocidad. Sin embargo, pude ver el contenido y cómo utilizarlos en la página semanal de la clase en Canvas.

Aplicar

Estas son las soluciones generadas para esta semana:

1. 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, por ejemplo:
 - Entre temperaturas de 0 y 25 grados, imprimir Temperatura Baja.
 - Entre temperaturas de 25 y 40 grados, imprimir Temperatura Agradable.
 - Más de 40 grados de temperatura, imprimir Temperatura Alta.



```
// C++ code
#include <Adafruit_LiquidCrystal.h>

Adafruit_LiquidCrystal lcd_1(0);

int valor = 0;
float centigrados = 0.0;

void setup()
{
  lcd_1.begin(20, 4);
  Serial.begin(9600);
}

void loop()
{
  valor = analogRead(0);
  centigrados = ((500.0 / 1023.0) * valor) - 50.0;
  Serial.print(centigrados);
  Serial.println(" C");
  if(centigrados >= 0 && centigrados <= 25){
    lcd_1.clear();
    lcd_1.print("Temperatura Baja");
  }else if(centigrados > 25 && centigrados <= 40){
    lcd_1.clear();
    lcd_1.setCursor(0,0);
    lcd_1.print("Temperatura");
    lcd_1.setCursor(0,1);
    lcd_1.print("Agradable");
  }else if(centigrados > 40){
    lcd_1.clear();
    lcd_1.print("Temperatura Alta");
  }
  delay(2000);
}
```

Auto-evaluación

Los resultados fueron positivos y correctos ya que se pudieron generar soluciones que satisfacen las especificaciones de esta semana.

Esto ayudará para la elaboración de soluciones en semanas próximas y poder cumplir con el indicador de la rúbrica.

Informar las referencias bibliográficas

Plataforma utilizada para la solución: <https://www.tinkercad.com/>

Semana 3

Especificaciones

Para esta semana, las especificaciones dadas por el profesor son las siguientes:

1. Presentar una solución que utiliza: un microcontrolador (Arduino o ESP32), un LED, una pantalla (LCD o OLED) y un sensor a elegir (temperatura, movimiento, ultrasonido o infrarrojo)

2. Identificar un código para el sensor (número ficticio)
3. Presentar la lectura de los datos del sensor en la pantalla y enviarlos a la base de datos en Firebase, con la fecha, código del sensor, valor de la lectura e información del estatus (caliente, frío o agradable)
4. Crear una condición para validar los valores, por ejemplo (crear una condición de análisis propia):
 - Si la temperatura es menor a 20 grados es frío
 - Si la temperatura está entre 20 y 35 grados es agradable
 - Si la temperatura es mayor a 35 grados es caliente
5. El LED debe parpadear por 5 segundos para informar que se tienen datos en transmisión.

Restricciones

Para esta semana, las restricciones presentadas por el profesor son las siguientes:

1. Utilizar C++ o Python
2. Utilizar Json con Firebase
3. Utilizar Arduino Uno o ESP32
4. Usar un LED, una pantalla (LCD o OLED) y un sensor a elegir
5. Enviar 4 datos: código del sensor, valor del sensor, fecha y estatus de la condición de análisis.

Investigar

En la actividad de esta semana, no fue necesaria la investigación de contenido. El material visto en clase fue suficiente para generar una solución a las especificaciones.

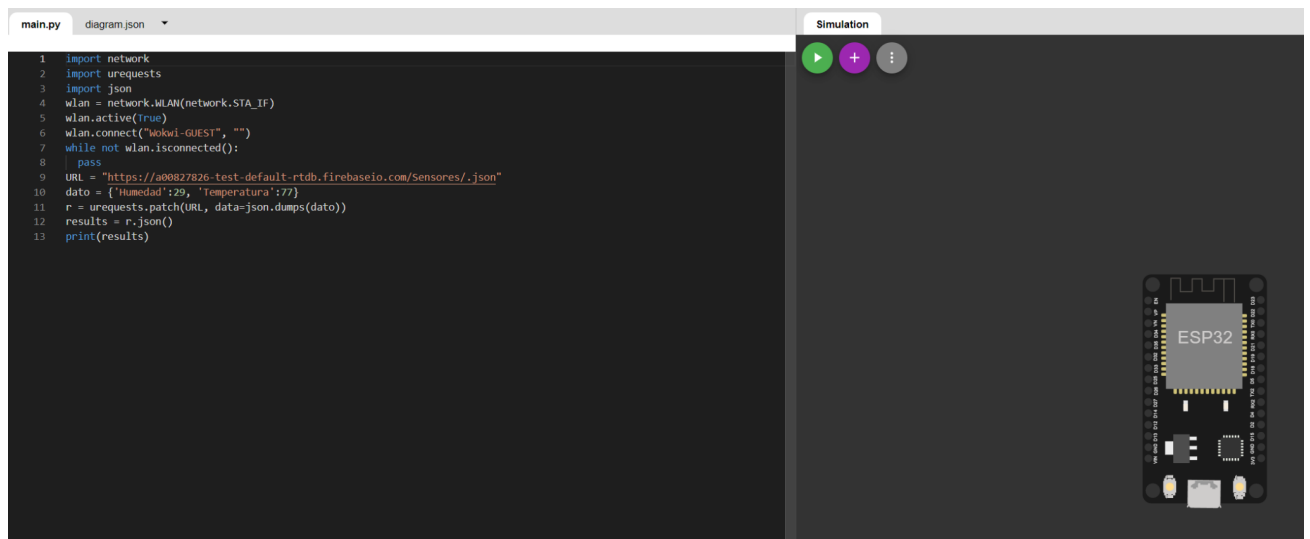
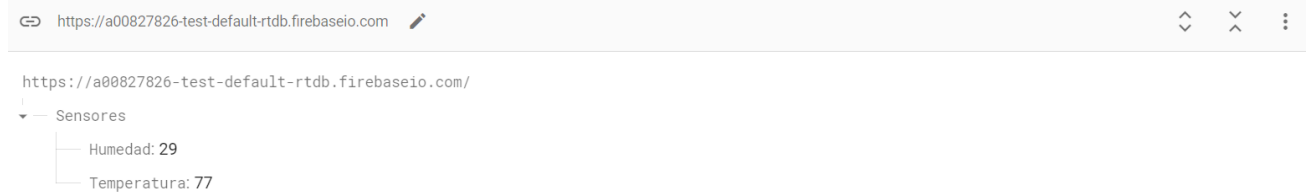
Experimentar

En esta semana, se pudo experimentar dentro de la clase realizando ejercicios de práctica, además de la explicación del profesor. Entre los ejercicios realizados estuvo:

- Introducción a la plataforma Wokwi
- Creación de un circuito en Wokwi
- Probar distintos componentes
- Conectar los datos con Firebase y observar cómo se actualiza en tiempo real

Interpretar

Aquí se muestran los resultados de la experimentación de esta semana. Principalmente, lo más importante fue la conexión realizada con éxito entre una base de datos en tiempo real usando Firebase y un circuito en Wokwi.



```
1 import network
2 import urequests
3 import json
4 wlan = network.WLAN(network.STA_IF)
5 wlan.active(True)
6 wlan.connect("Wokwi-GUEST", "")
7 while not wlan.isconnected():
8     pass
9 URL = "https://a00827826-test-default-rtdb.firebaseio.com/Sensores/.json"
10 dato = {'Humedad':29, 'Temperatura':77}
11 r = urequests.patch(URL, data=json.dumps(dato))
12 results = r.json()
13 print(results)
```

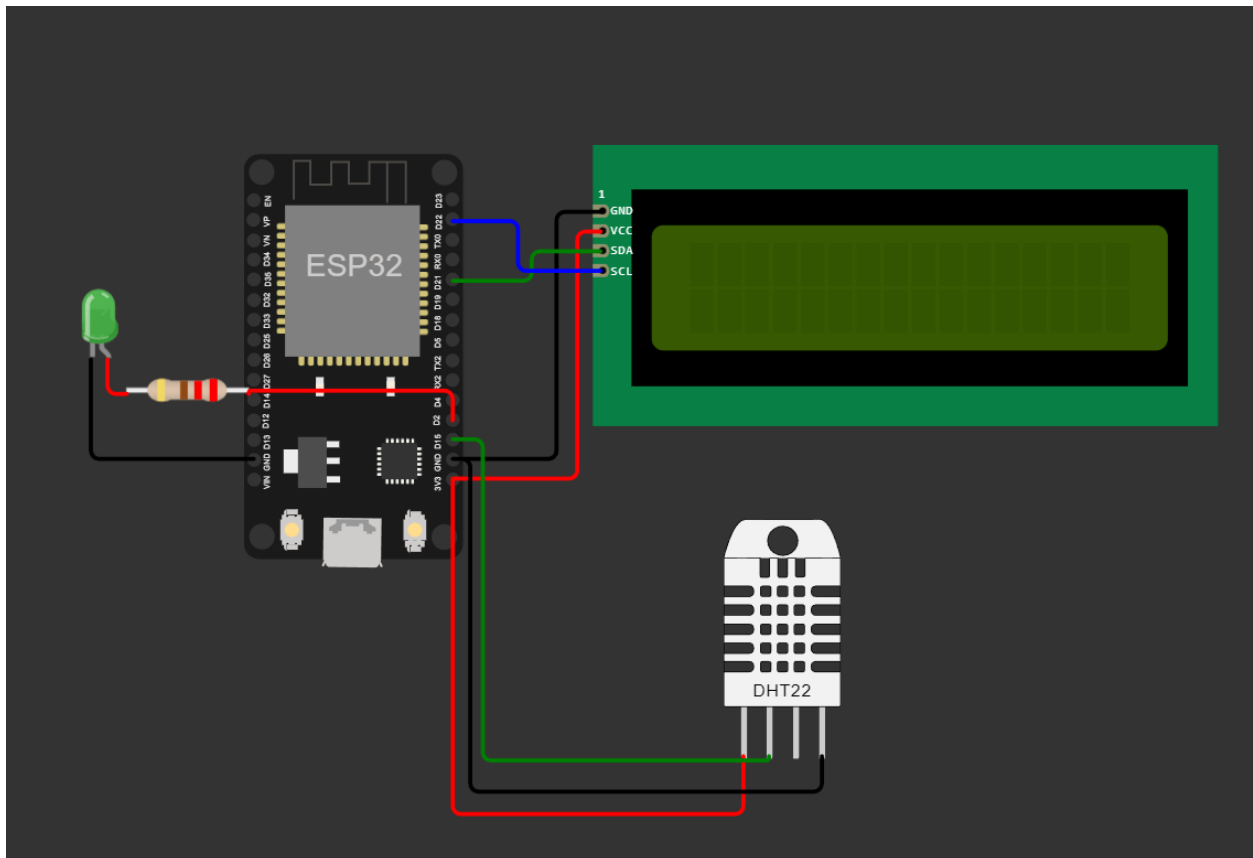

Esto evidentemente fue muy útil para poder realizar la actividad semanal.

Aplicar

Estas son las soluciones generadas para esta semana:

1. Presentar una solución que utiliza: un microcontrolador (Arduino o ESP32), un LED, una pantalla (LCD o OLED) y un sensor a elegir (temperatura, movimiento, ultrasonido o infrarrojo)

Se decidió optar por un microcontrolador ESP32, un LED verde, una pantalla LCD I2C y un sensor DHT22 de temperatura y humedad:

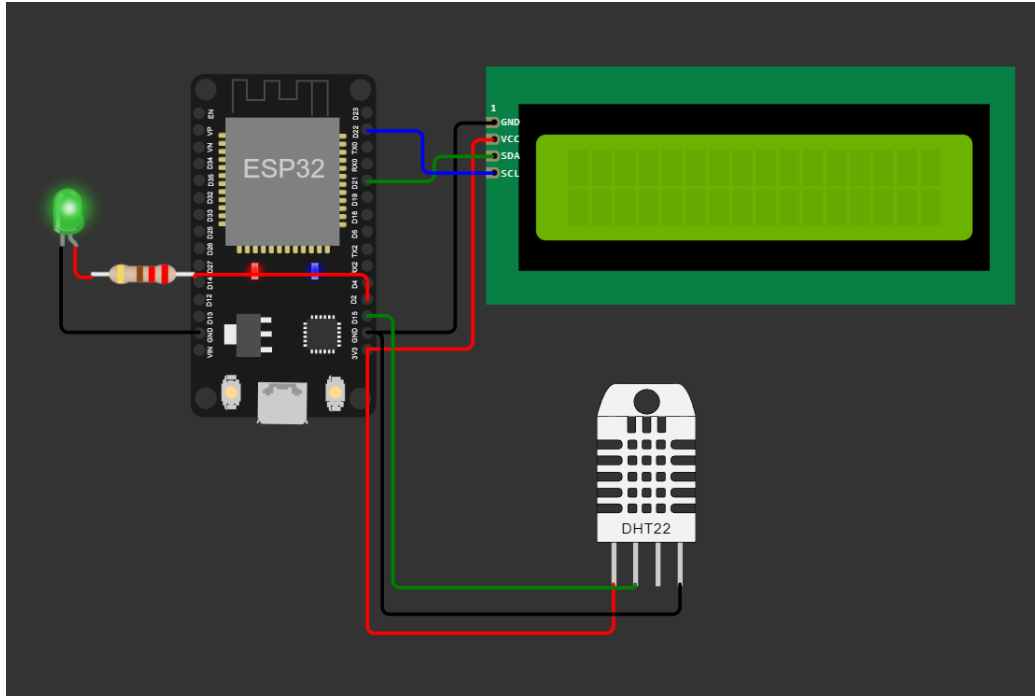


2. Identificar un código para el sensor (número ficticio)

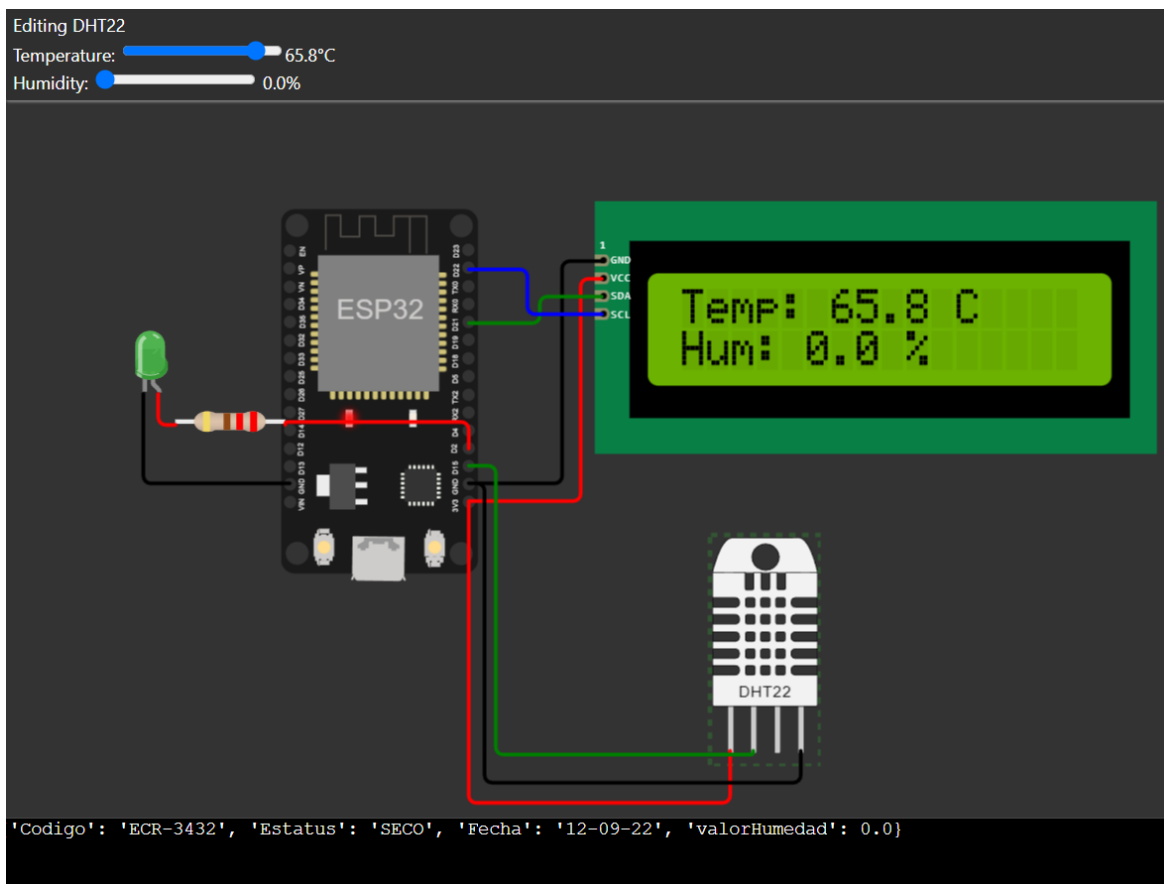
El código para el sensor es **ECR-3432**, fue elegido de manera arbitraria.

3. Presentar la lectura de los datos del sensor en la pantalla y enviarlos a la base de datos en Firebase, con la fecha, código del sensor, valor de la lectura e información del estatus

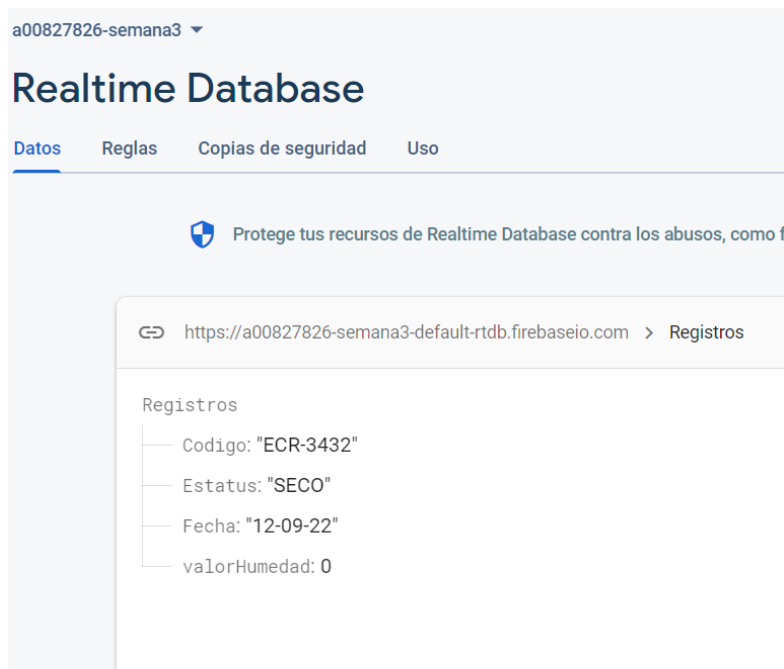
Al correr el código, se puede ver lo siguiente:



El LED cuando hay una transferencia de datos parpadea por cinco segundos, para después dar paso a la impresión de datos:



En la pantalla LCD se imprimen los valores de temperatura y humedad que tiene el sensor detectados. De igual manera, en la consola se ve la respuesta que regresa Firebase al mandarle los datos.



4. Crear una condición para validar los valores:

- Si la humedad es mayor al 75%, se cataloga como HÚMEDO
- Si la humedad es mayor al 25% y menor a 75%, se cataloga como AGRADABLE
- Si la humedad es menor a 25%, se cataloga como SECO

Dependiendo de la humedad se establece el estatus:



Auto-evaluación

Los resultados fueron positivos y correctos ya que se pudieron generar soluciones que satisfacen las especificaciones de esta semana.

Esto ayudará para la elaboración de soluciones en semanas próximas y poder cumplir con el indicador de la rúbrica.

En adición, el único detalle que no terminé de arreglar fue que quería hacer que la fecha se consiguiera de manera automática con una librería pero no pude incluirla e importarla. Es por esta razón que la mandé como string.

Informar las referencias bibliográficas

Plataforma utilizada para la solución (link solución): <https://wokwi.com/projects/342500541072409171>

Firebase: <https://console.firebase.google.com/>

Semana 4

Especificaciones

Para esta semana, las especificaciones dadas por el profesor son las siguientes:

1. Presentar una propuesta para la utilización de servicios Cloud para el envío y recibimiento de datos con Firebase.
2. Integrar a la solución de la semana 3, utilizando los mismos componentes y reglas para validación de datos.
3. Presentar en Cloud el envío y recibimiento 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 vía Cloud.

Restricciones

Para esta semana, las restricciones presentadas por el profesor son las siguientes:

1. Utilizar AWS IoT core o Arduino Cloud o Google Cloud o Azure IoT Central o Python Web Server
2. Utilizar Arduino o ESP32 como microcontrolador
3. Utilizar C++ o Micro Python
4. Presentar una gráfica para presentación de los datos
5. Presentar una operación de control de su microcontrolador por su proyecto Cloud

Investigar

En la actividad de esta semana, no fue necesaria la investigación de contenido. El material visto en clase fue suficiente para generar una solución a las especificaciones.

Experimentar

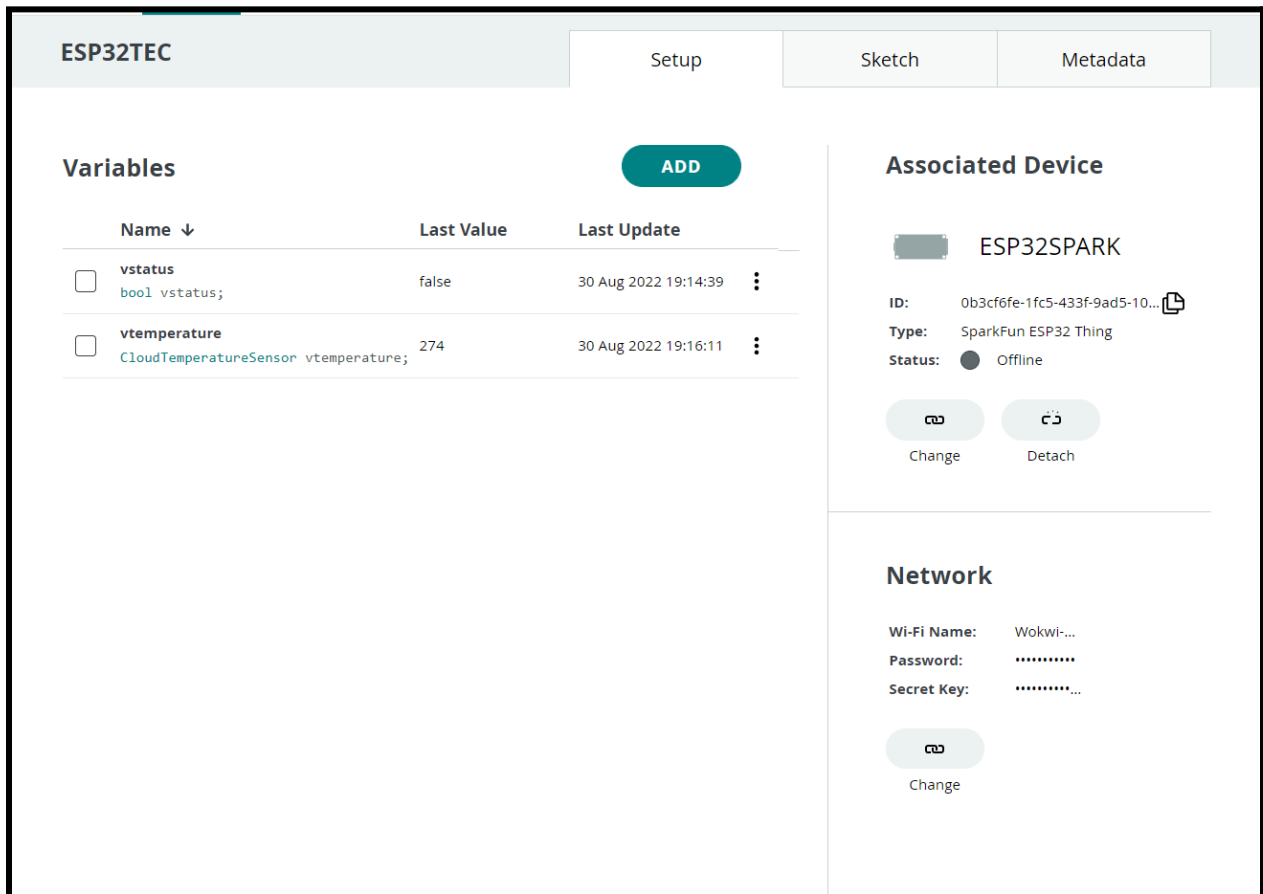
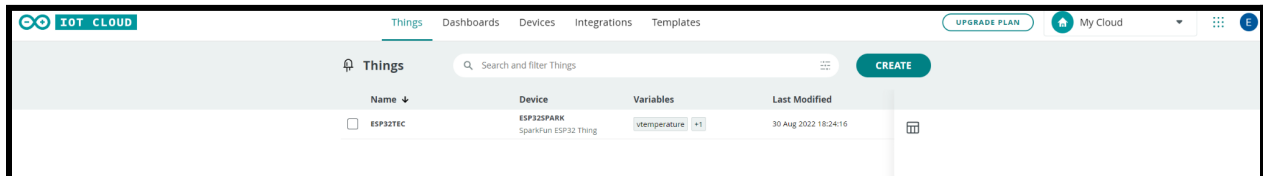
En esta semana, se pudo experimentar dentro de la clase realizando ejercicios de práctica, además de la explicación del profesor. Entre los ejercicios realizados estuvo:

- Introducción a la plataforma Arduino IOT Cloud
- Creación de un circuito en Wokwi conectado con Arduino IOT Cloud

- Creación de un Dashboard en Arduino IOT Cloud

Interpretar

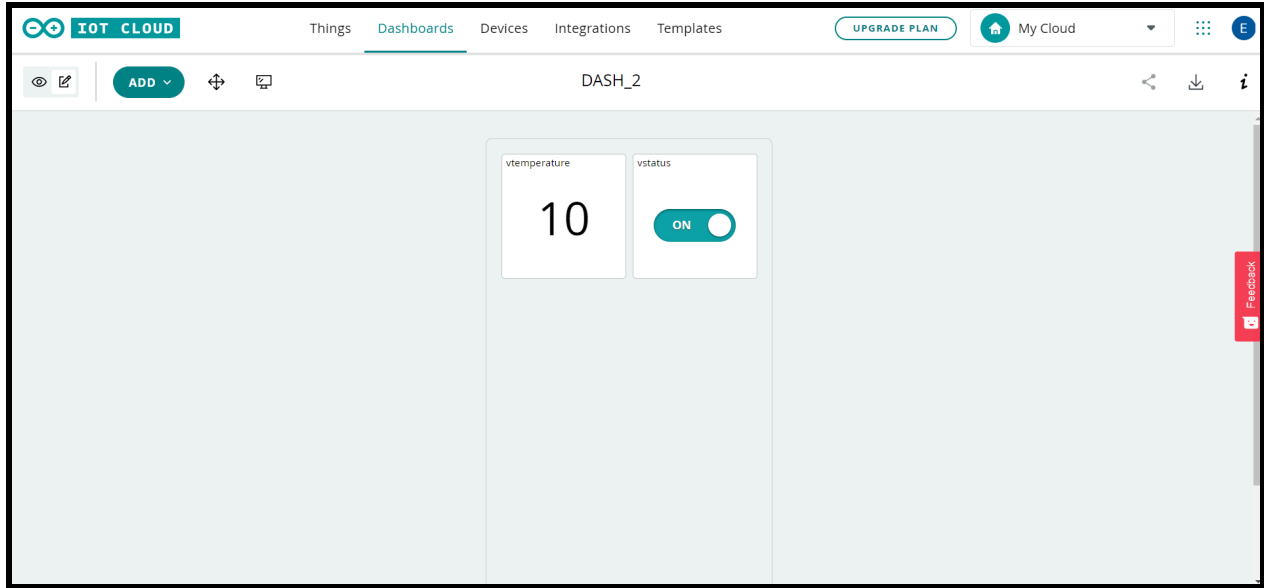
Aquí se muestran los resultados de la experimentación de esta semana. Principalmente, lo más importante fue la conexión realizada con éxito entre el Dashboard de Arduino IOT Cloud y el circuito en Wokwi:



Simulation

A detailed simulation of an ESP32 microcontroller module. The module is shown with its various components: a central ESP32 chip, a red LED, a white LED, a USB-C port, and two gold-colored pins at the bottom. The pins are labeled with their respective functions: VIN, GND, D13, D12, D14, D27, D26, D25, D23, D33, D35, D34, V+, EN, and D22, D21, D20, D19, D18, D16, D15, D4, D3, D2, D1, D0, TX, RX.

```
MQTT Broker: mqttt-up.1ot.arduino.cc:8884
WiFi status ESP: 255
Connection to "Wokwi-GUEST" failed
Retrying in "500" milliseconds
Connected to "Wokwi-GUEST"
Connected to Arduino IoT Cloud
Thing ID: 742af5cb-f4b0-4012-91aa-96daf96f3c5c
OFF
ArduinoIoTCLOUDTCP::handle_Disconnect MQTT client connection lost
Disconnected from Arduino IoT Cloud
Connected to Arduino IoT Cloud
Thing ID: 742af5cb-f4b0-4012-91aa-96daf96f3c5c
OFF
ON
ON
OFF
```



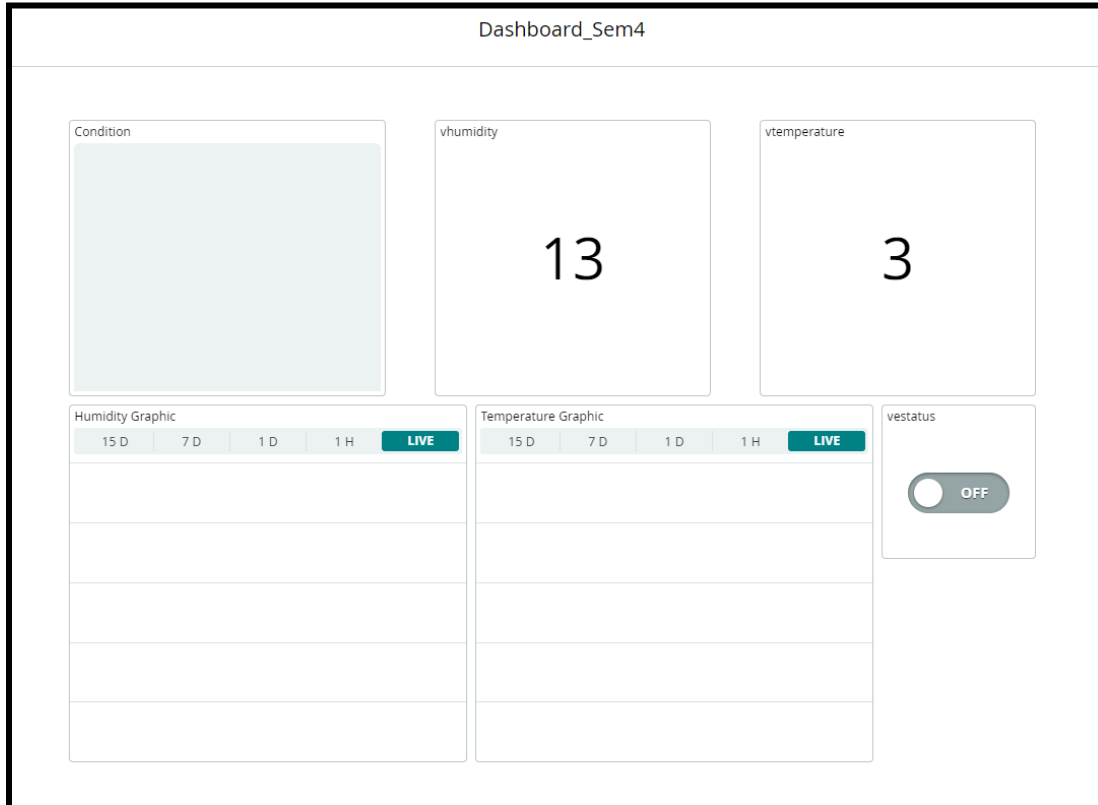
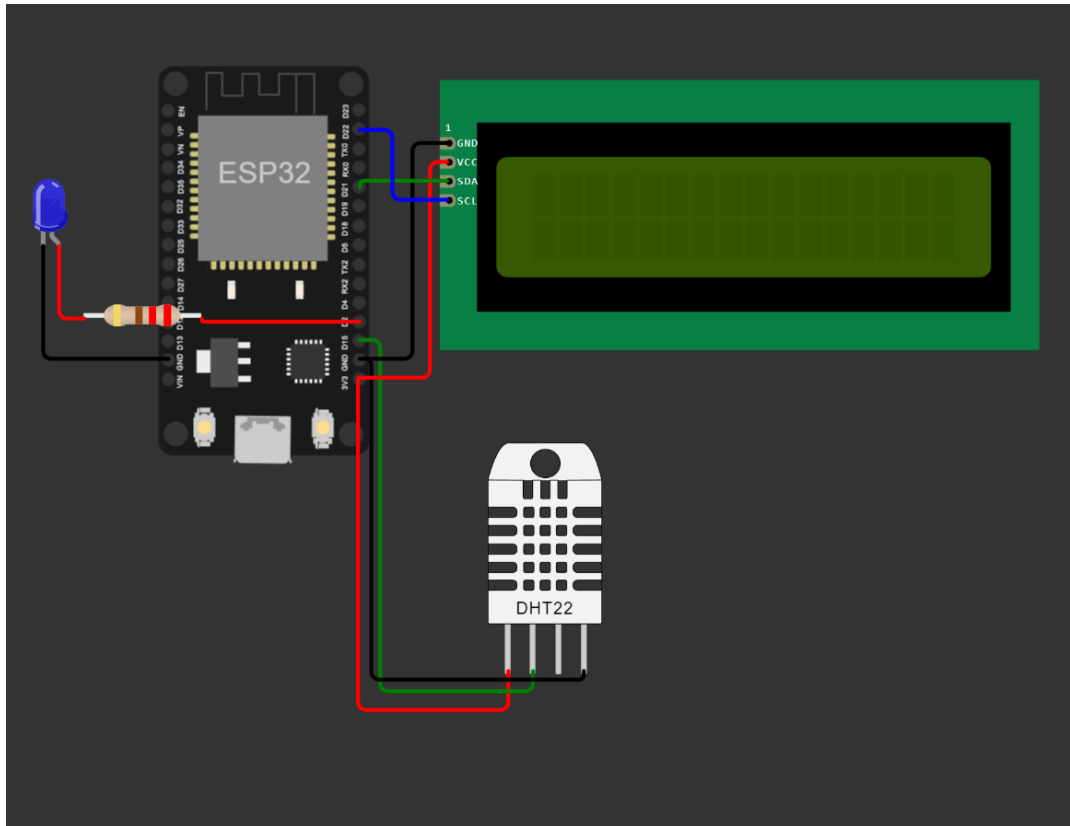
Con un código simple, se demuestra que el microcontrolador puede ser manejado desde el dashboard en Arduino IOT Cloud.

Aplicar

Estas son las soluciones generadas para esta semana:

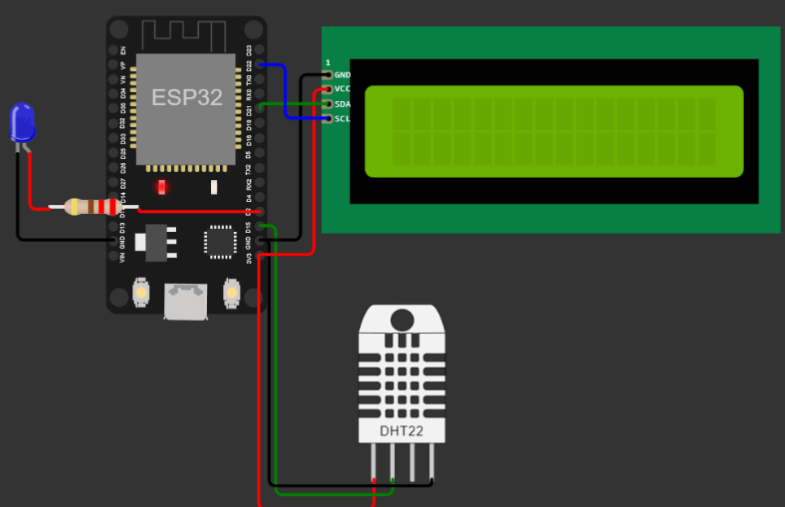
1. Presentar una propuesta para la utilización de servicios Cloud para el envío y recibimiento de datos con Firebase.
2. Integrar a la solución de la semana 3, utilizando los mismos componentes y reglas para validación de datos.
3. Presentar en Cloud el envío y recibimiento de datos, también una presentación 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 vía Cloud.

Para la solución, se plantea el uso de la plataforma Arduino IOT Cloud. El circuito será manejado desde un dashboard, y cada vez que se quiera tomar un registro o conocer la temperatura y humedad del lugar, se deberá activar un interruptor. Cabe mencionar que las reglas y los componentes se mantienen iguales. Cuando se inicie la ejecución, un foco LED tendrá un parpadeo en 5 segundos:

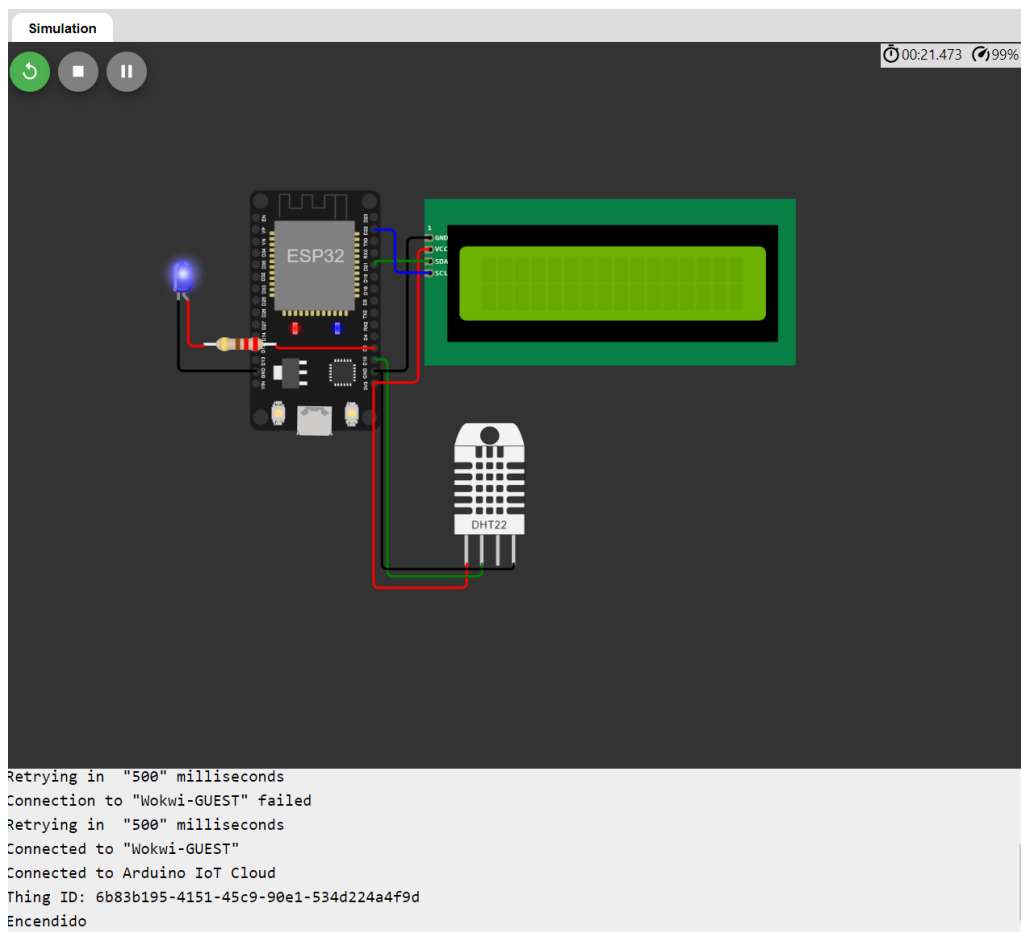
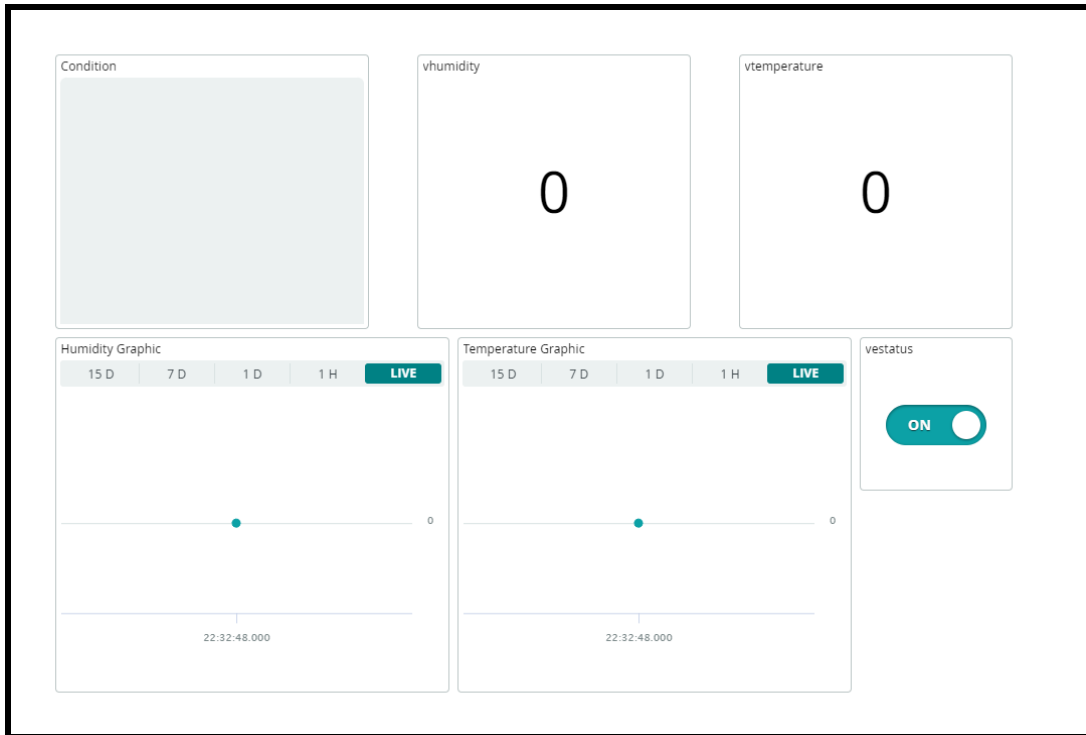


Simulation

00:21.038 85%



```
***** Arduino IoT Cloud - configuration info *****
Device ID: 0b3cf6fe-1fc5-433f-9ad5-10b8b613e7a5
MQTT Broker: mqtt5-up.iot.arduino.cc:8884
WiFi status ESP: 255
Connection to "Wokwi-GUEST" failed
Retrying in "500" milliseconds
Connection to "Wokwi-GUEST" failed
Retrying in "500" milliseconds
Connected to "Wokwi-GUEST"
Connected to Arduino IoT Cloud
Thing ID: [REDACTED] f9d
```



Simulation

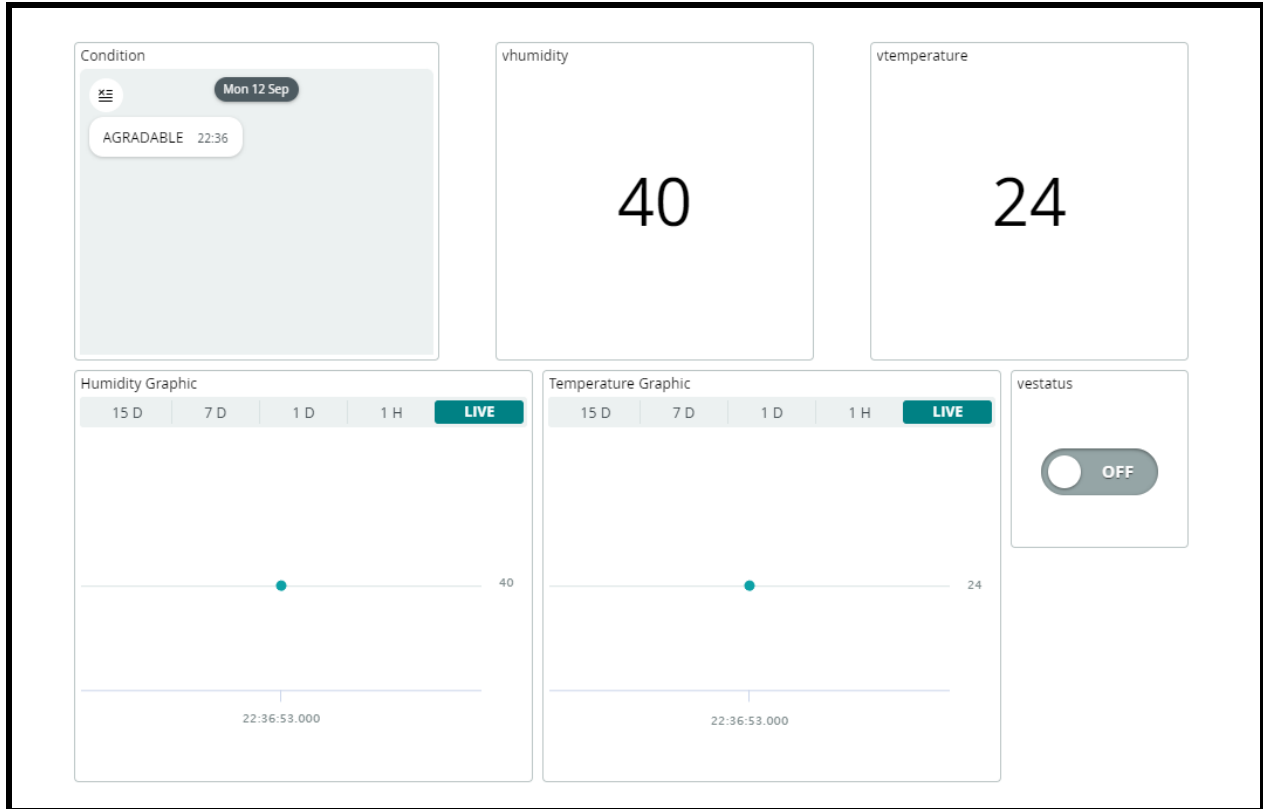
00:31.945 69%

ESP32

Temp: 24.00 C
Hum: 40.00 %

DHT22

Connection to "Wokwi-GUEST" failed
Retrying in "500" milliseconds
Connected to "Wokwi-GUEST"
Connected to Arduino IoT Cloud
Thing ID: 6b83b195-4151-45c9-90e1-534d224a4f9d
Encendido
AGRADABLE



Como se puede observar, los datos son transmitidos correctamente.

Auto-evaluación

Los resultados fueron positivos y correctos ya que se pudieron generar soluciones que satisfacen las especificaciones de esta semana.

Esto ayudará para la elaboración de soluciones en semanas próximas y poder cumplir con el indicador de la rúbrica.

En adición, este es el último paso importante, para la semana cinco es una simple mejora al dashboard que será explicada a continuación. Considero que la actividad de esta semana se realizó con éxito pues podría estar manejando el controlador desde un celular y recibir la información en el momento de la temperatura y humedad. Simplemente tengo que accionar el LED.

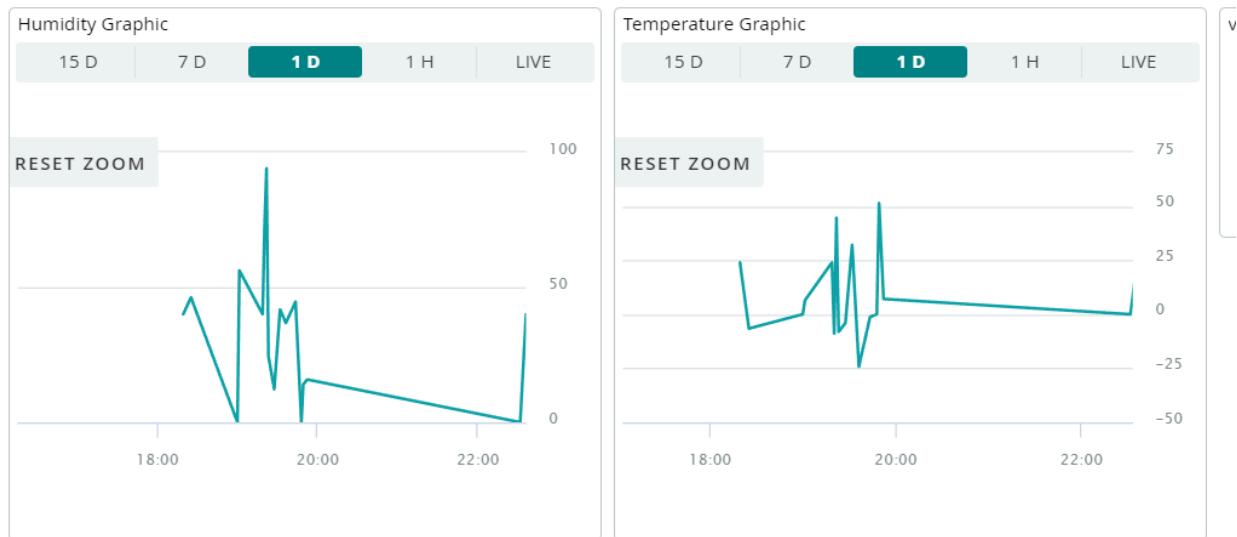
Informar las referencias bibliográficas

El dashboard en Arduino no lo puedo compartir porque la cuenta es gratuita. El link del wokwi es <https://wokwi.com/projects/342622645292892754>

Semana 5

Para la semana 5, se nos indicó que solamente será necesaria la implementación de un gráfico de decisiones y tendencias para la solución de la semana 4.

En sí, para este caso se agregaron un par de gráficos en el dashboard para poder observar el flujo de dos datos: humedad y temperatura.



Resulta útil pues nos puede apoyar en saber cómo ha cambiado el comportamiento de la variable en vivo, ayer, o hace una semana. También nos ayudaría a conocer la naturaleza de los datos y tener la posibilidad de realizar predicciones por ejemplo. En este caso, las gráficas presentan formas atípicas ya que se introdujeron mediciones de prueba.

En cuanto a la toma de decisiones, solamente se tiene este interruptor, que es el que activa la transmisión de datos, la pantalla y el LED. Si este elemento no es seleccionado, solamente estará la pantalla encendida, pero no mostrará datos. Decidí que cuando terminase de mandar datos, apagara el interruptor, de esta manera si el usuario quiere volver a realizar una medición, debe encenderlo.

