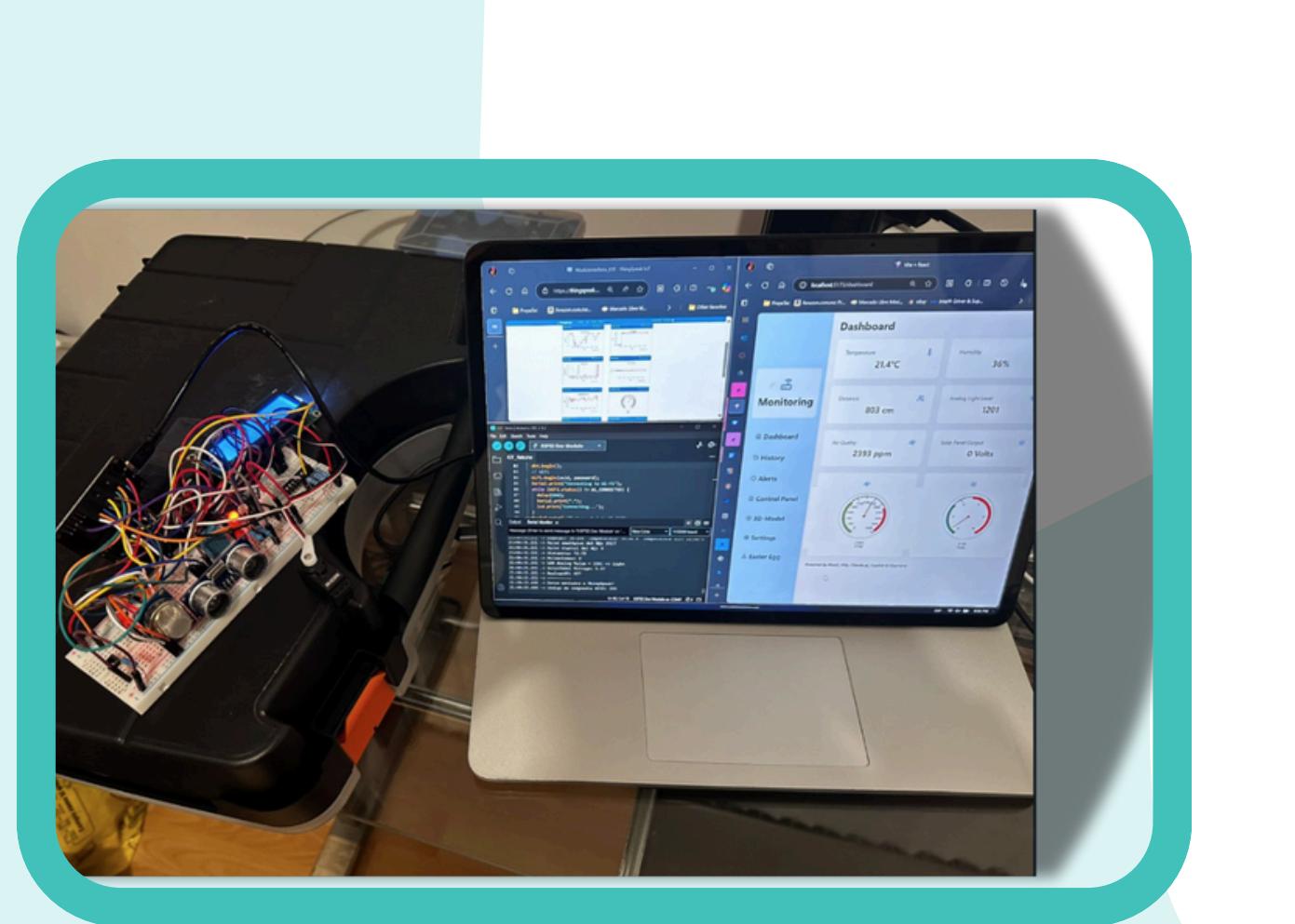




**PRESENTACIÓN**

# Reto: Almacén Inteligente

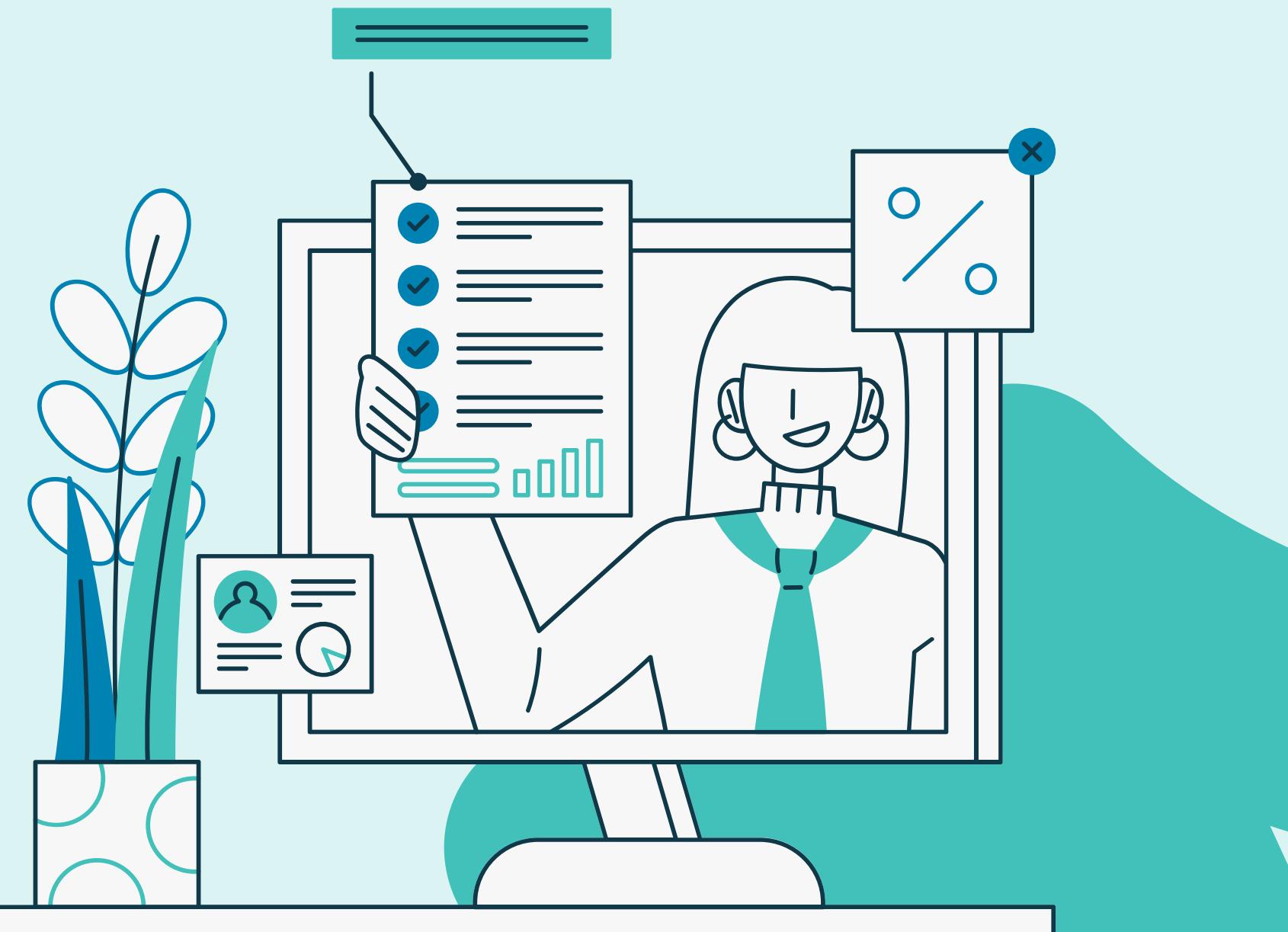
Implementación de Internet de las Cosas  
(Grupo 503)

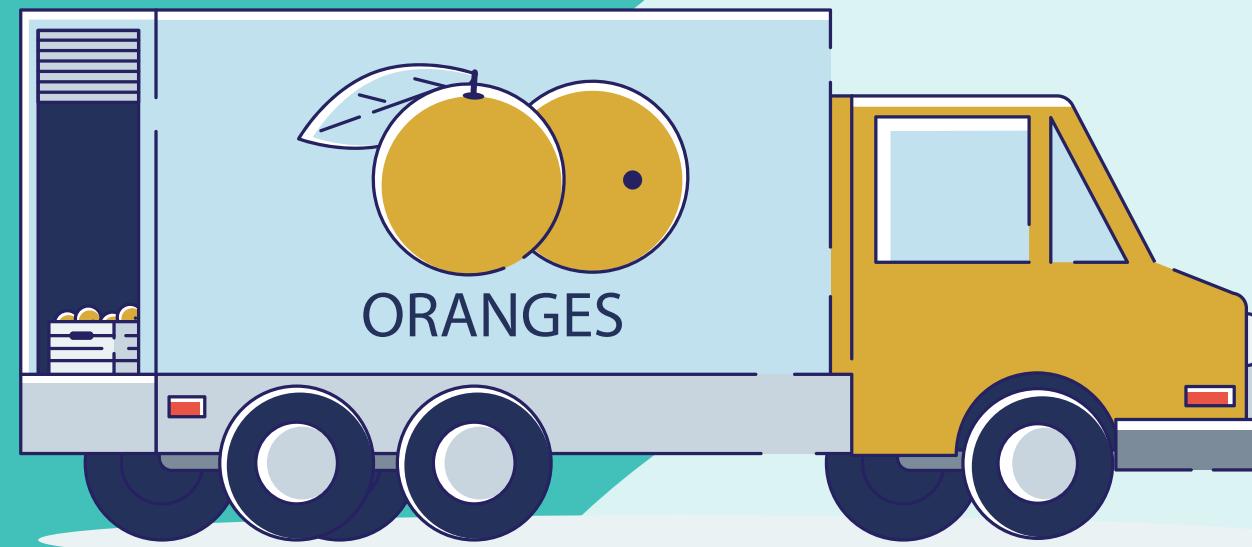


Diego de la Vega Saishio A01420632  
Héctor Lugo Gabino A01029811  
Mauricio Monroy González A01029647  
Luis Veledíaz Flores A01029829

# Contenidos

- 01.** Introducción.
- 02.** Sensores Utilizados.
- 03.** Periféricos Utilizados.
- 04.** Fases del Desarrollo.
- 06.** Datos en ThingSpeak.
- 07.** Cifras Graficadas.
- 08.** Conclusión final.





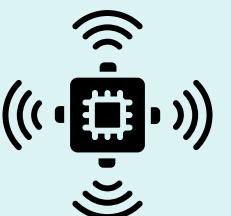
Alrededor del 20% de los alimentos refrigerados se pierden anualmente debido a fallos en el mantenimiento de temperaturas óptimas durante su almacenamiento y transporte.

Sensores IoT permiten monitorear estas condiciones en tiempo real, reduciendo significativamente estas pérdidas al alertar sobre cambios de temperatura que podrían comprometer la calidad del producto

Tech, N. (2024, 26 septiembre). IoT en la industria alimentaria: Usos y alcances esenciales - NC Tech. NC Tech. <https://nctech.mx/nc-tech-insights/iot-en-la-industria-alimentaria/>

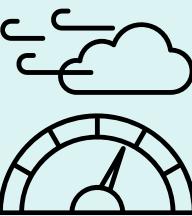


## Sensores utilizados:



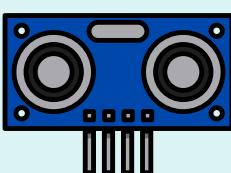
### Sensor DHT11

- Tomar datos de temperatura y humedad.



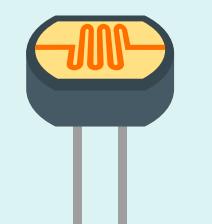
### Sensor de Calidad del Aire MQ-135

- Tomar datos analógicos sobre partículas por millón en el ambiente.



### Sensor Ultrasónico

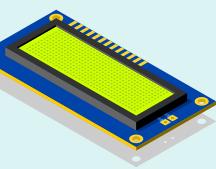
- Para monitorear quien entra y sale del almacén por medio del movimiento.



### Light Dependent Resistor (fotoresistencia)

- Para monitorear la intensidad de la luz en el entorno.

# Periféricos utilizados:



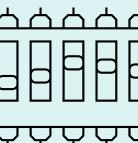
## LCD Display

- Para mostrar la información de las medidas que se deseen consultar en tiempo real.



## Push Button, LED y Resistencias

- Para tomar pulsaciones y mostrar información con LEDs



## Dip Switch (Pseudo Decodificador 3:9)

- Para controlar qué información se despliega en el LCD Display según lo que el usuario quiera ver.



## ServoMotor

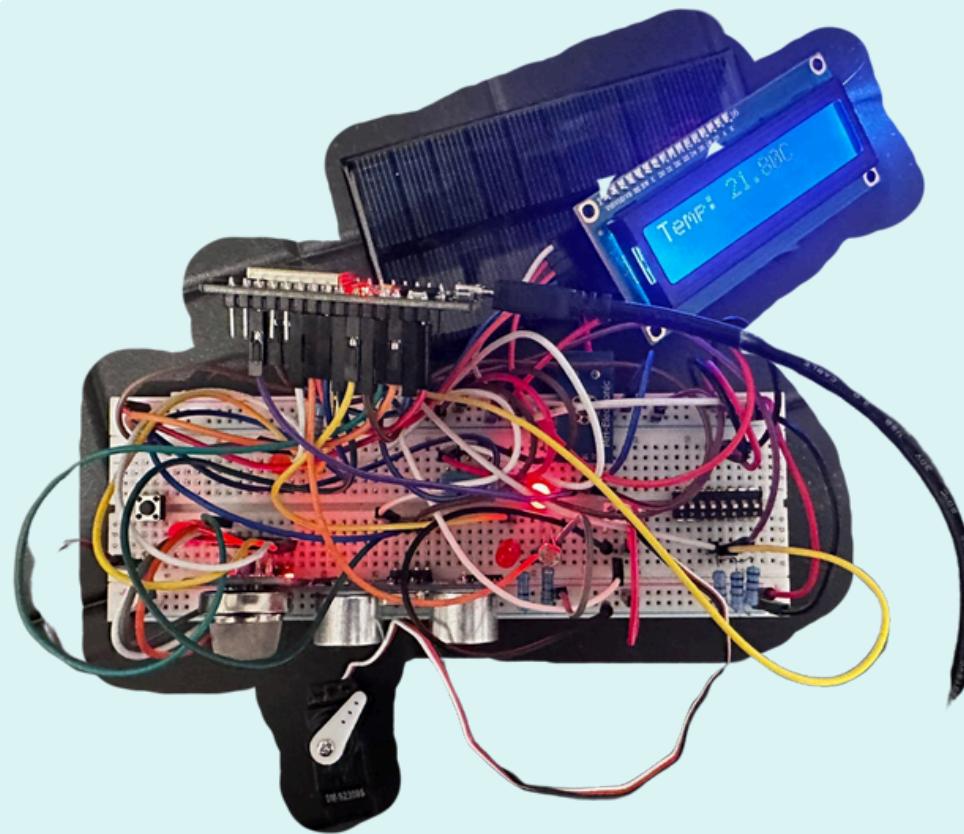
- Para simular una escotilla de un sistema de ventilamiento dependiendo del estado de la calidad del aire.



## Panel Solar

- Simulación de generación de energía renovable

# Fases del desarrollo del sistema de IoT



FASE 1

## Diseño de Circuito

Aprovechando avances en clases de Hardware y las prácticas, se combinaron los conocimientos en un prototipo funcional.

FASE 2

## Configuración de ThingSpeak

Activando 7 campos y Widgets, se preparó la exportación de gráficos y visualizaciones, al igual que preparando colección de datos en un CSV.

Channel Settings

Percentage Complete	30%	
Channel ID	2739575	
Name	MedicionesReto_IOT	
Description		
Field 1	Temperatura	<input checked="" type="checkbox"/>
Field 2	Humedad	<input checked="" type="checkbox"/>
Field 3	Calidad de Aire	<input checked="" type="checkbox"/>
Field 4	Distancia	<input checked="" type="checkbox"/>
Field 5	Pulsaciones	<input checked="" type="checkbox"/>
Field 6	Light Analog Value	<input checked="" type="checkbox"/>
Field 7	Solar Panel Output	<input checked="" type="checkbox"/>
Field 8		<input type="checkbox"/>

# Snippets

```
#include <HTTPClient.h>
#include <DHT.h>
#include <LiquidCrystal_I2C.h>
#include <ESP32Servo.h>

#define DHTTYPE DHT11
#define TRIG_PIN 26
#define ECHO_PIN 25
#define MqAnalogPin 32
#define MqDigitalPin 19
#define BUTTON_PIN 13
#define LIGHT_SENSOR 33
```

```
// Selectores
int s1 = 35;
int s2 = 34;
int s3 = 4;
int ledAir = 12;
int ledDHT = 14;
int ledButt = 23;
int ledDist = 27;

Servo myServo; // Servo object

// Configuración Wi-Fi
const char* ssid = "Mau";
const char* password = "carspotter1";
/*
const char* ssid = "InMorris-WiFi";
const char* password = "Umpalumpa.149";
*/
// Configuración de Thingspeak
String apiKey = "858RKIG8JVX5DXJJQ";
const char* server = "http://api.thingspeak.com";

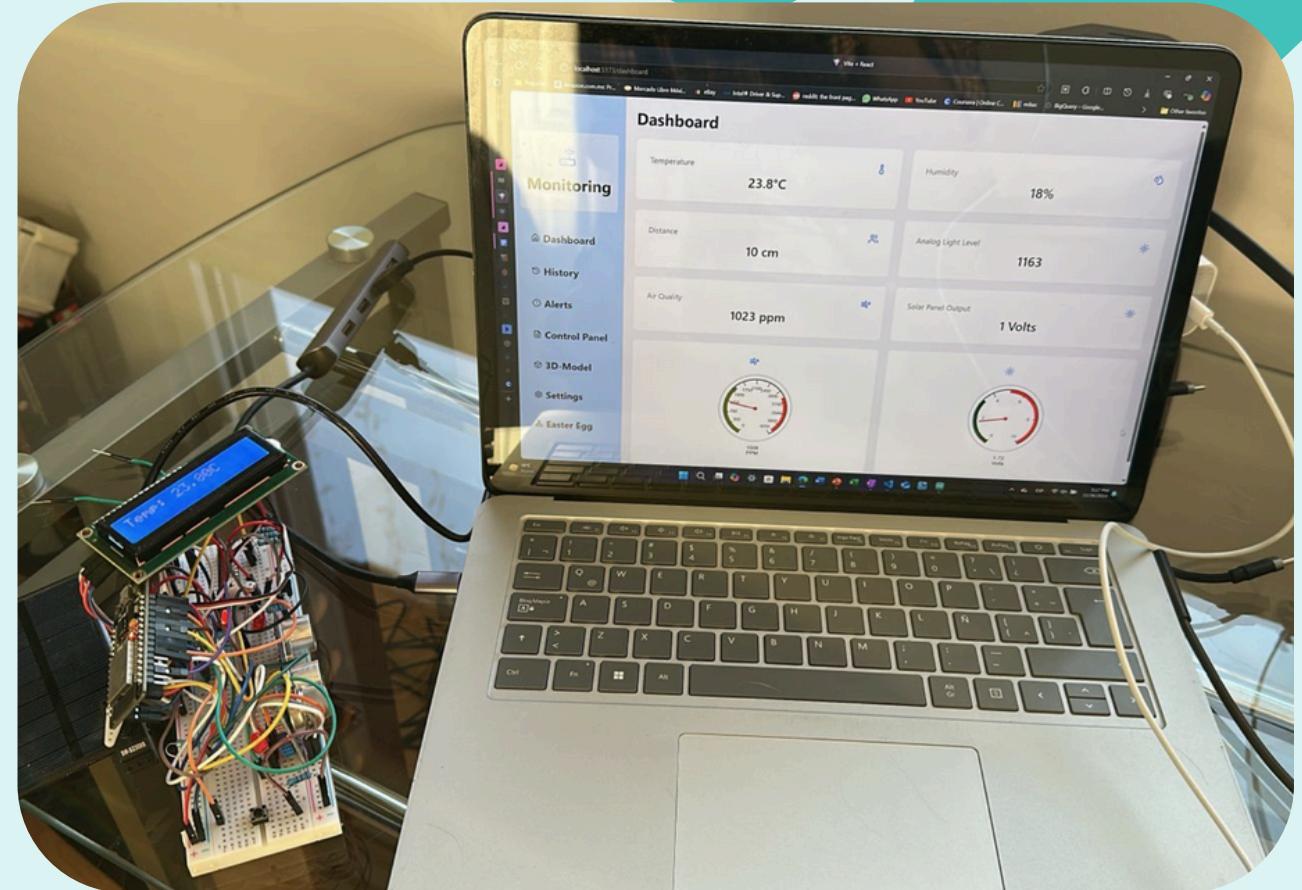
// Sensores
DHT dht(18, DHTTYPE);
float duration_us, distance_cm;

volatile int count = 0; // Must be volatile for use in ISR
// Debouncing variables
volatile unsigned long lastDebounceTime = 0;
unsigned long debounceDelay = 50; // milliseconds
void IRAM_ATTR handleButtonPress() {
    unsigned long currentTime = millis();
    digitalWrite(ledButt, HIGH);
    if ((currentTime - lastDebounceTime) > debounceDelay) {
        count++;
        digitalWrite(ledButt, HIGH);
        lastDebounceTime = currentTime;
    }
}
```

```
void setup() {
    // LCD
    lcd.init();
    lcd.backlight();
    // Serial
    Serial.begin(115200);
    // PinModes
    pinMode(TRIG_PIN, OUTPUT);
    pinMode(ECHO_PIN, INPUT);
    pinMode(MqDigitalPin, INPUT);
    pinMode(BUTTON_PIN, INPUT_PULLUP); // Pullup
    pinMode(s1, INPUT);
    pinMode(s2, INPUT);
    pinMode(s3, INPUT);
    pinMode(ledAir, OUTPUT);
    pinMode(ledDist, OUTPUT);
    pinMode(ledDHT, OUTPUT);
    pinMode(ledButt, OUTPUT);
    analogSetAttenuation(ADC_11db);
    // DHT
    dht.begin();
    // WiFi
    WiFi.begin(ssid, password);
    Serial.print("Connecting to Wi-Fi");
    while (WiFi.status() != WL_CONNECTED) {
        delay(500);
        Serial.print(".");
        lcd.print("Connecting...");
    }
    Serial.println("\nConnected to Wi-Fi");
    Serial.print("IP Address: ");
    Serial.println(WiFi.localIP());
    delay(2000);
    // Interrupt
    attachInterrupt(digitalPinToInterrupt(BUTTON_PIN), handleButtonPress, FALLING);
    // Servo
    myServo.attach(15); // Pin 15
}
```

# Fases del desarrollo del sistema de IoT

```
JavaScript > React > dashboard2iot > src > App.jsx > [o] App
23 import Alerts from './components/Alerts';
24 import ControlPanel from './components/ControlPanel';
25 import Tetris from './components/Tetris';
26 import './index.css';
27
28 const App = () => {
29   const [sensorData, setSensorData] = useState({
30     temperature: 0,
31     humidity: 0,
32     airQuality: 0,
33     distance: 0,
34     lightLevel: 0,
35     solarPanel: 0
36   });
37   const [historicalData, setHistoricalData] = useState({
38     labels: [],
39     temperature: [],
40     humidity: [],
41     airQuality: [],
42     distance: [],
43     lightLevel: [],
44     solarPanel: []
45   });
46
47   return (
48     <div>
49       <Alerts sensorData={sensorData} />
50       <ControlPanel />
51       <Tetris />
52       <Dashboard sensorData={sensorData} historicalData={historicalData} />
53     </div>
54   );
55 }
56
57 export default App;
```



FASE 3

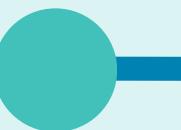


## Desarrollo de App

Se desarrolló una interfaz con el uso de Javascript React, con las siguientes secciones:

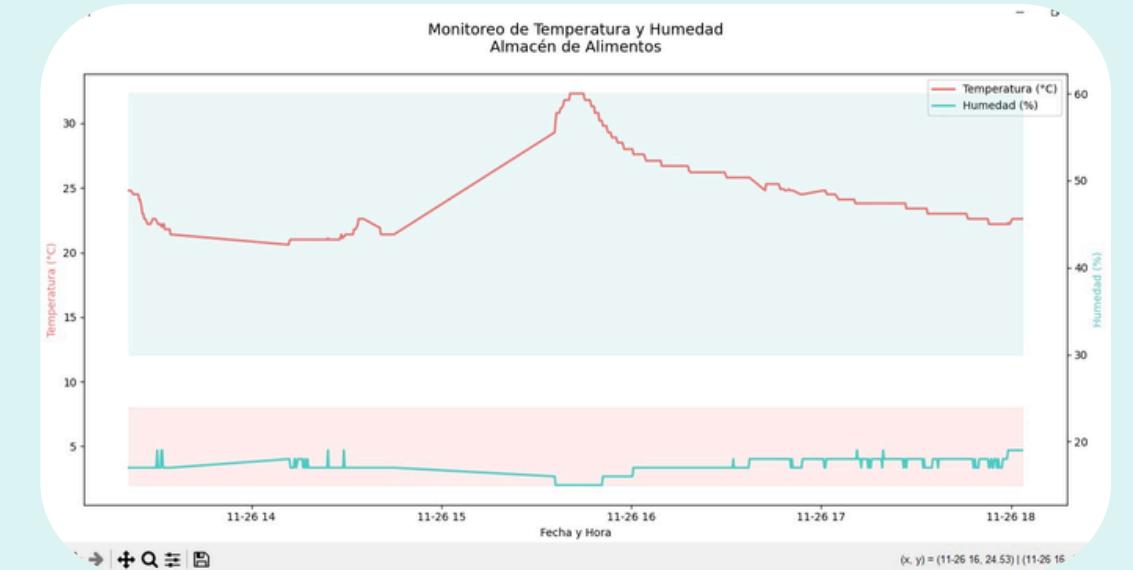
- Dashboard
- Historial
- Alertas
- Panel de control
- Modelo 3D
- Configuración
- Easter Egg

FASE 4



## Testing

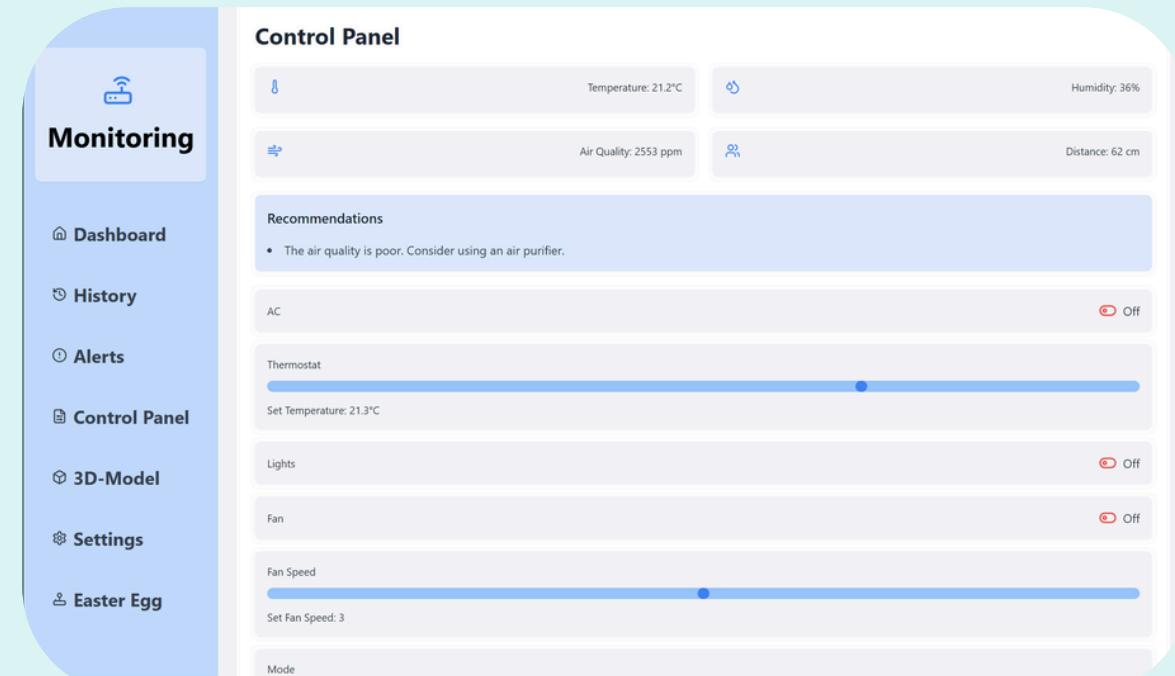
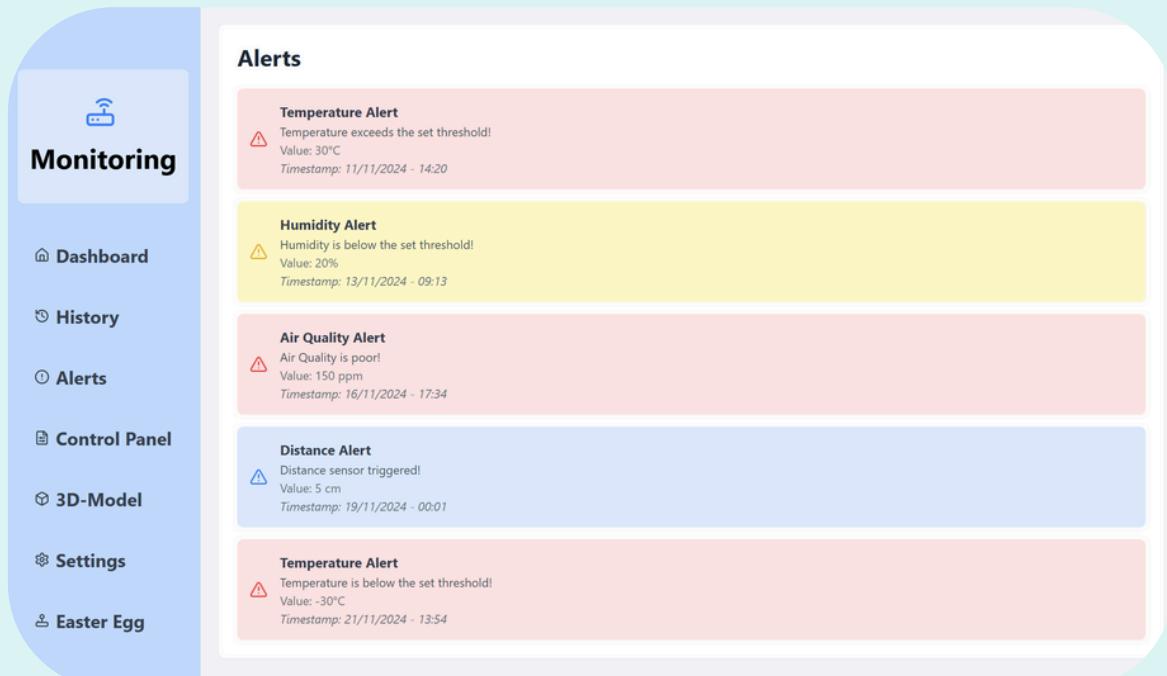
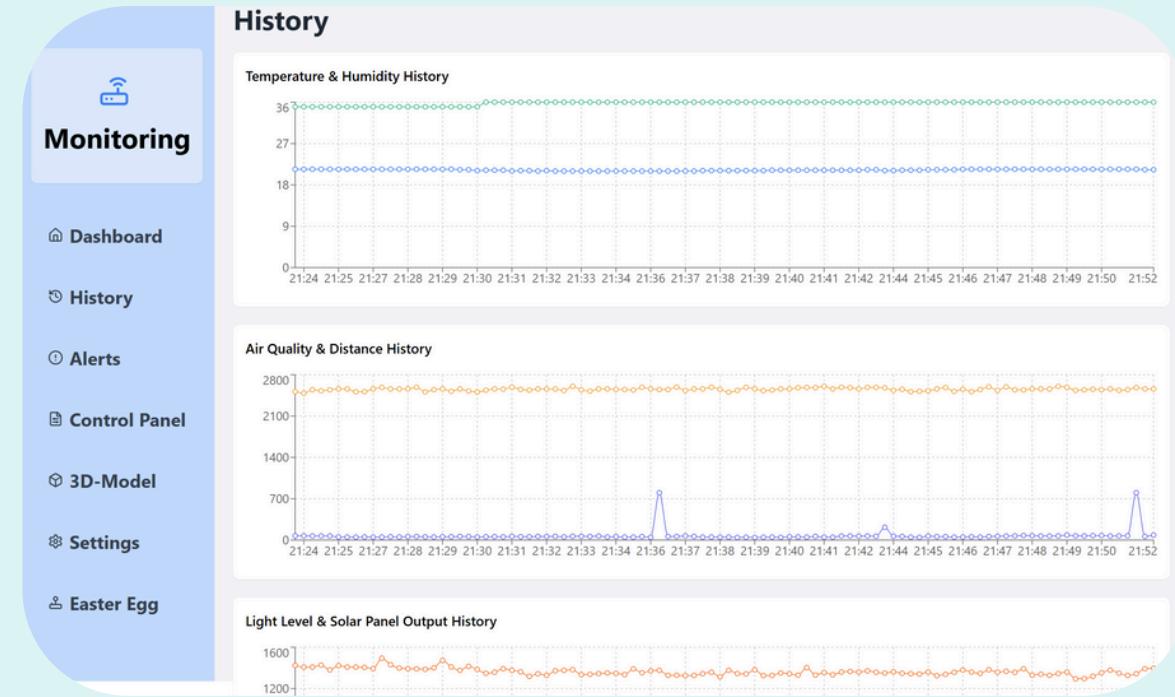
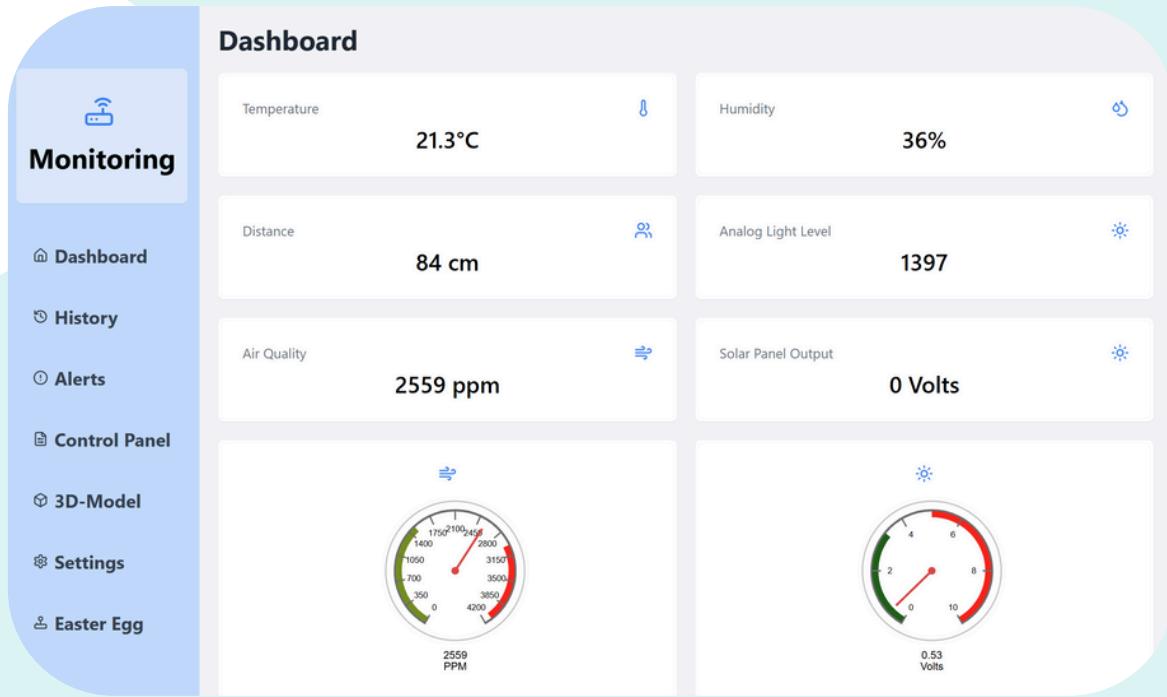
Activando 7 campos y Widgets, se preparó la exportación de gráficos y visualizaciones, al igual que preparando colección de datos en un CSV



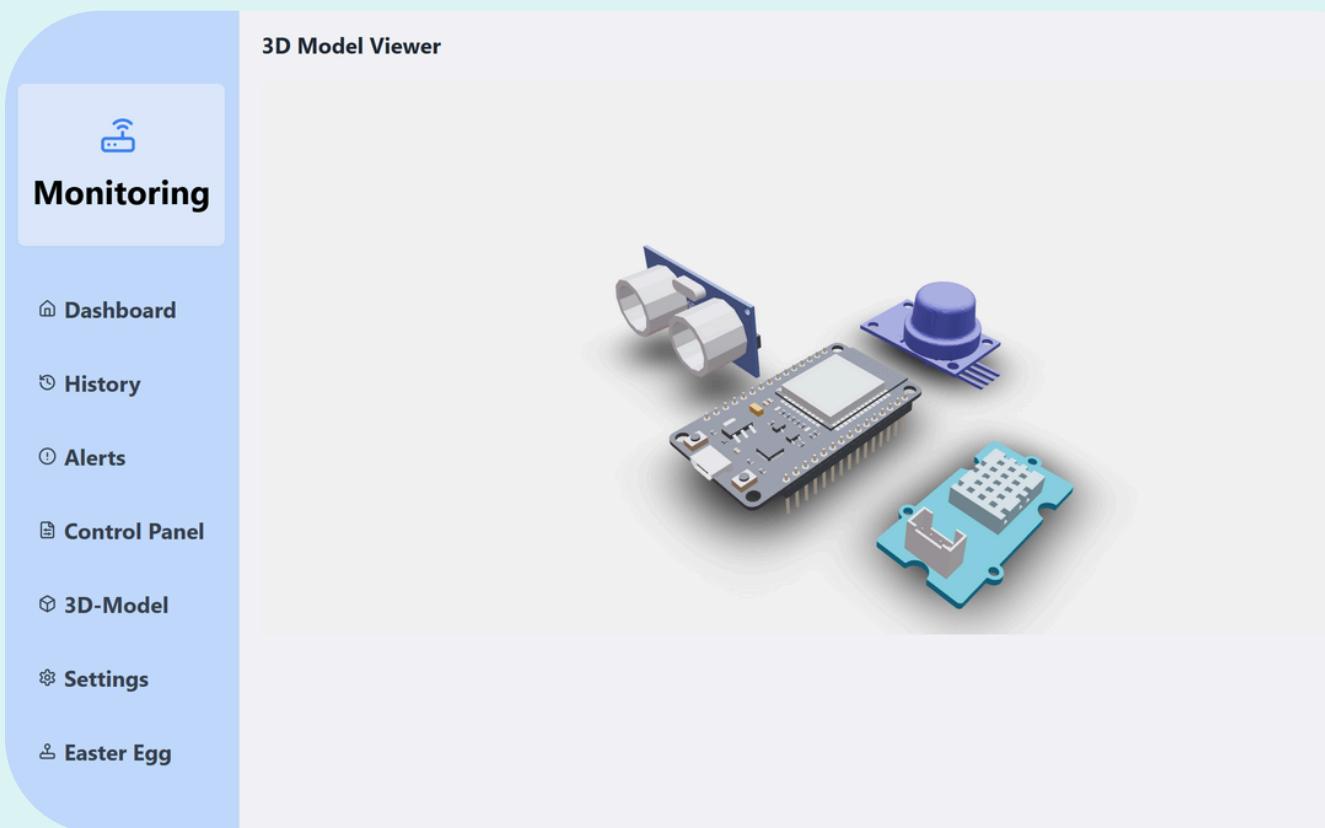


# Monitoring

# Interfaz



# Interfaz

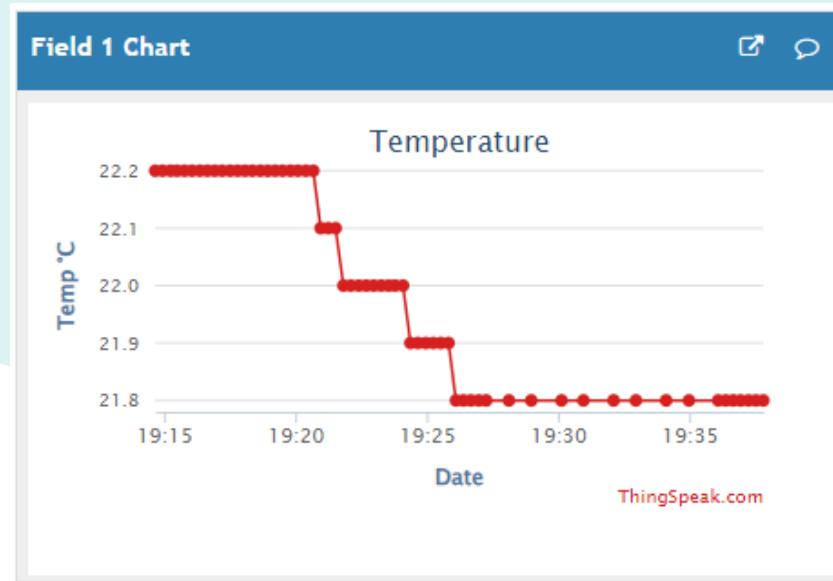


This screenshot shows the Settings interface. It features a sidebar with Monitoring, Dashboard, History, Alerts, Control Panel, 3D-Model, Settings, and Easter Egg. The main content area contains several slider controls for monitoring thresholds:

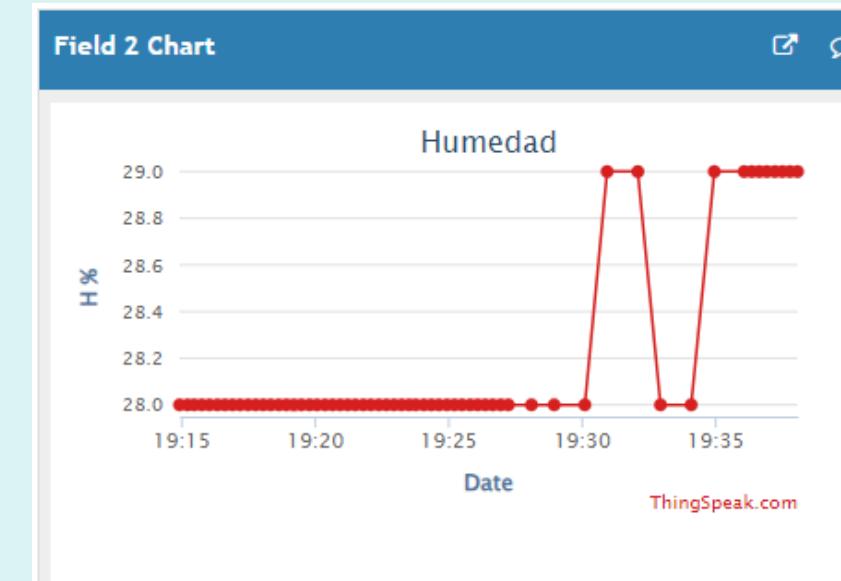
- Alarm Sound Level:** Sound: 10
- Temperature Threshold:** Min Temperature Threshold: 0°C, Max Temperature Threshold: 40°C
- Humidity Threshold:** Min Humidity Threshold: 20%, Max Humidity Threshold: 60%
- Air Quality Threshold:** Air Quality Threshold: 100 ppm

# ThingSpeak

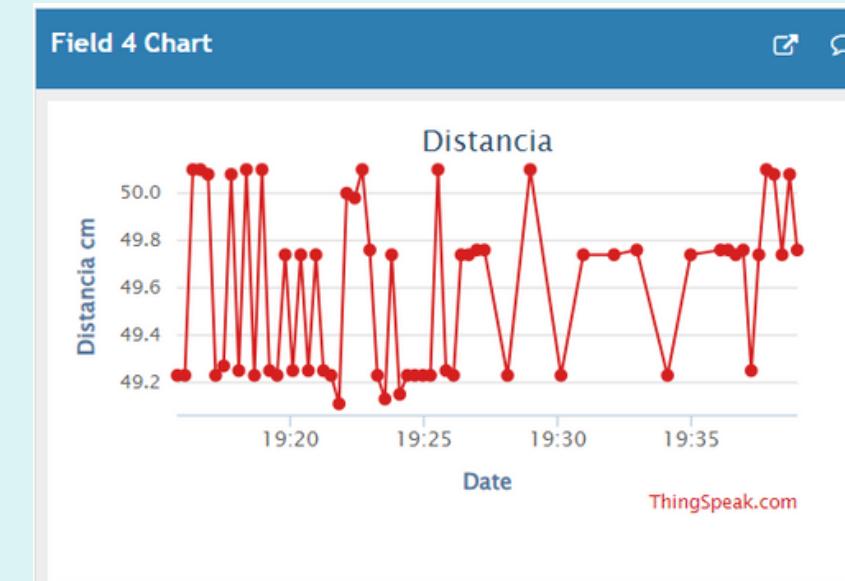
Datos recopilados: 24,408 (corte en 26/11/2024-21:31)



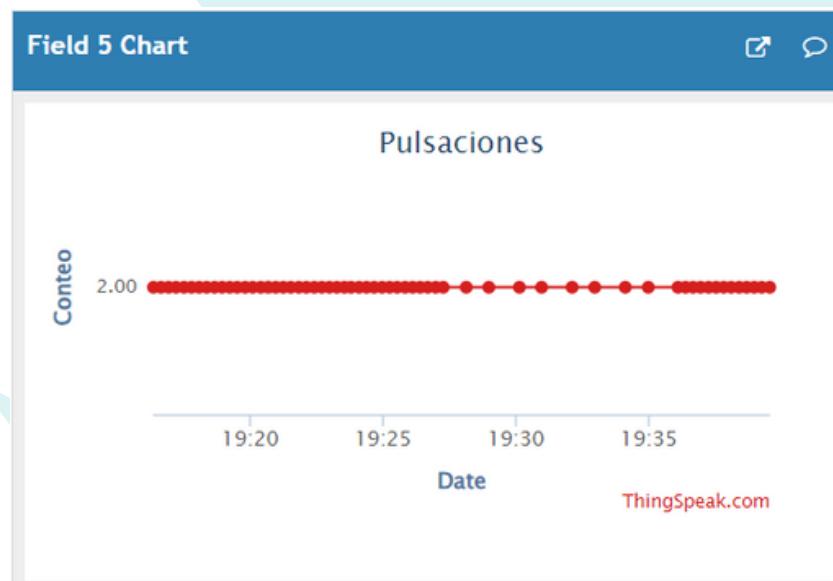
Temperatura



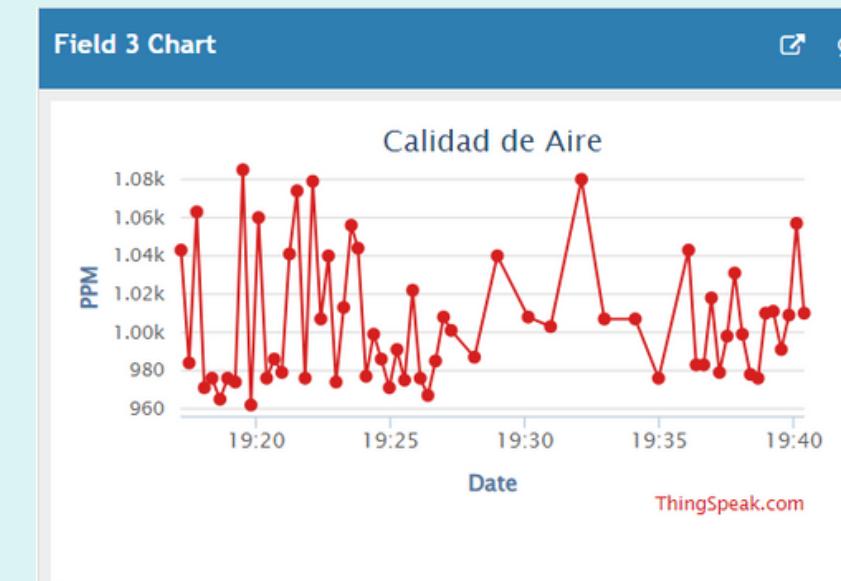
Humedad



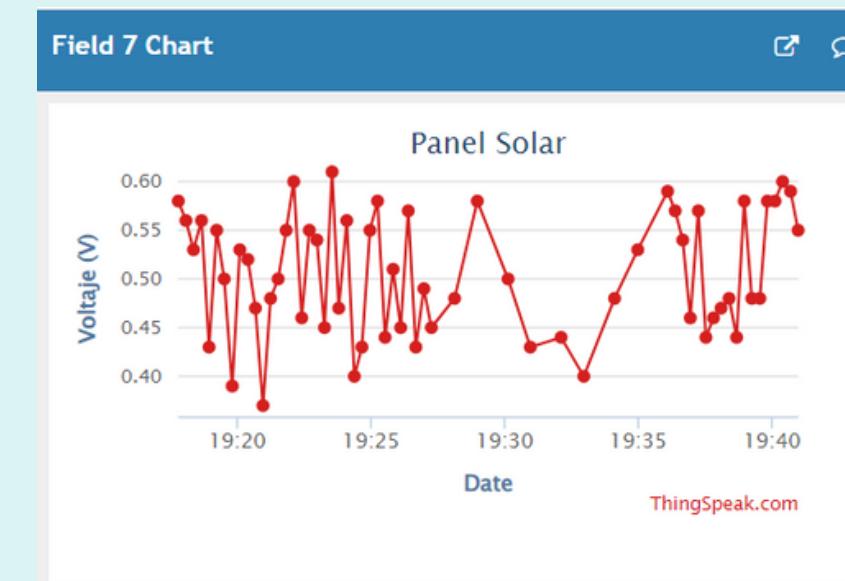
Distancia



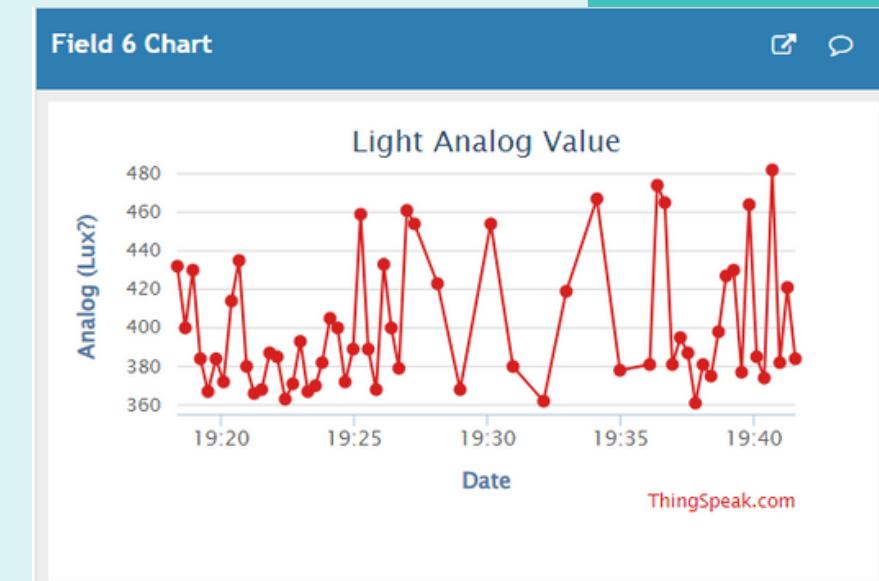
Pulsaciones



Calidad del aire

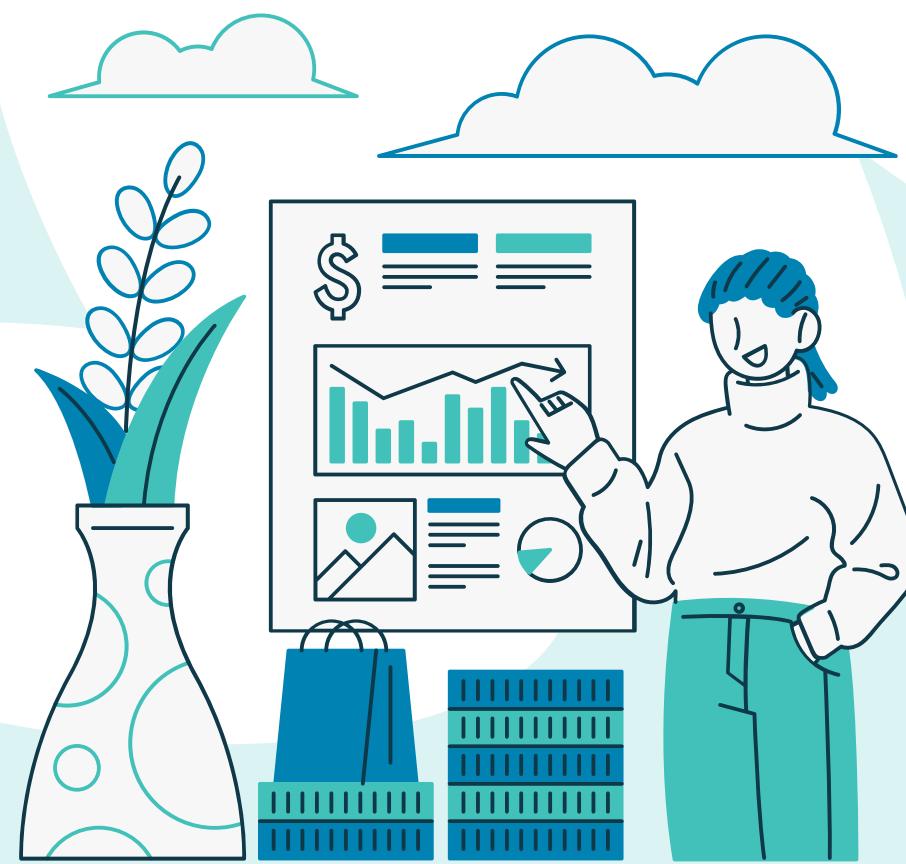


Panel Solar



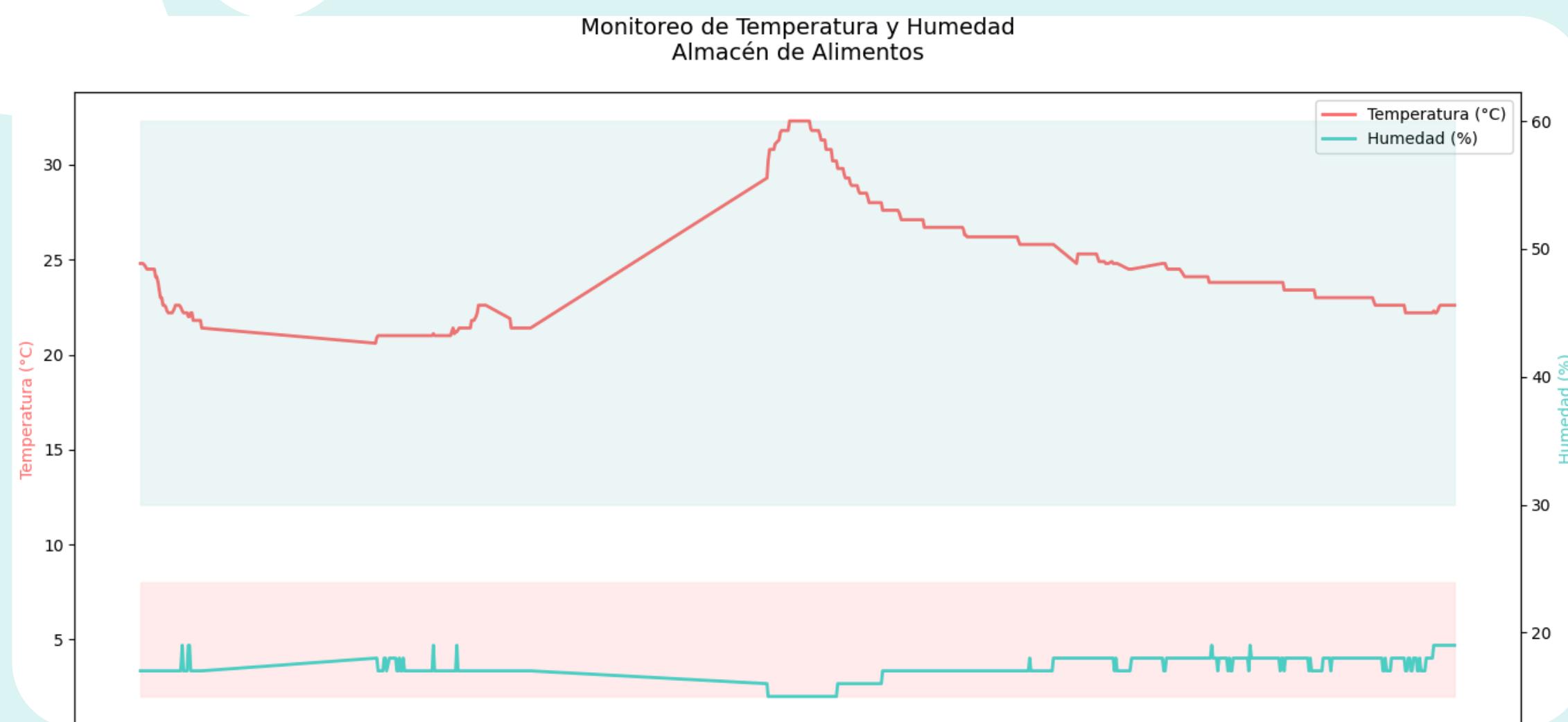
Valor Análogo de la luz





# Datos y cifras: Un día ordinario de Caso de Uso

# CSV - Temperatura y Humedad



Temperatura promedio: 24.6°C

Temperatura máxima: 32.3°C

Temperatura mínima: 20.6°C

Humedad promedio: 17.2%

Humedad máxima: 19.0%

Humedad mínima: 15.0%

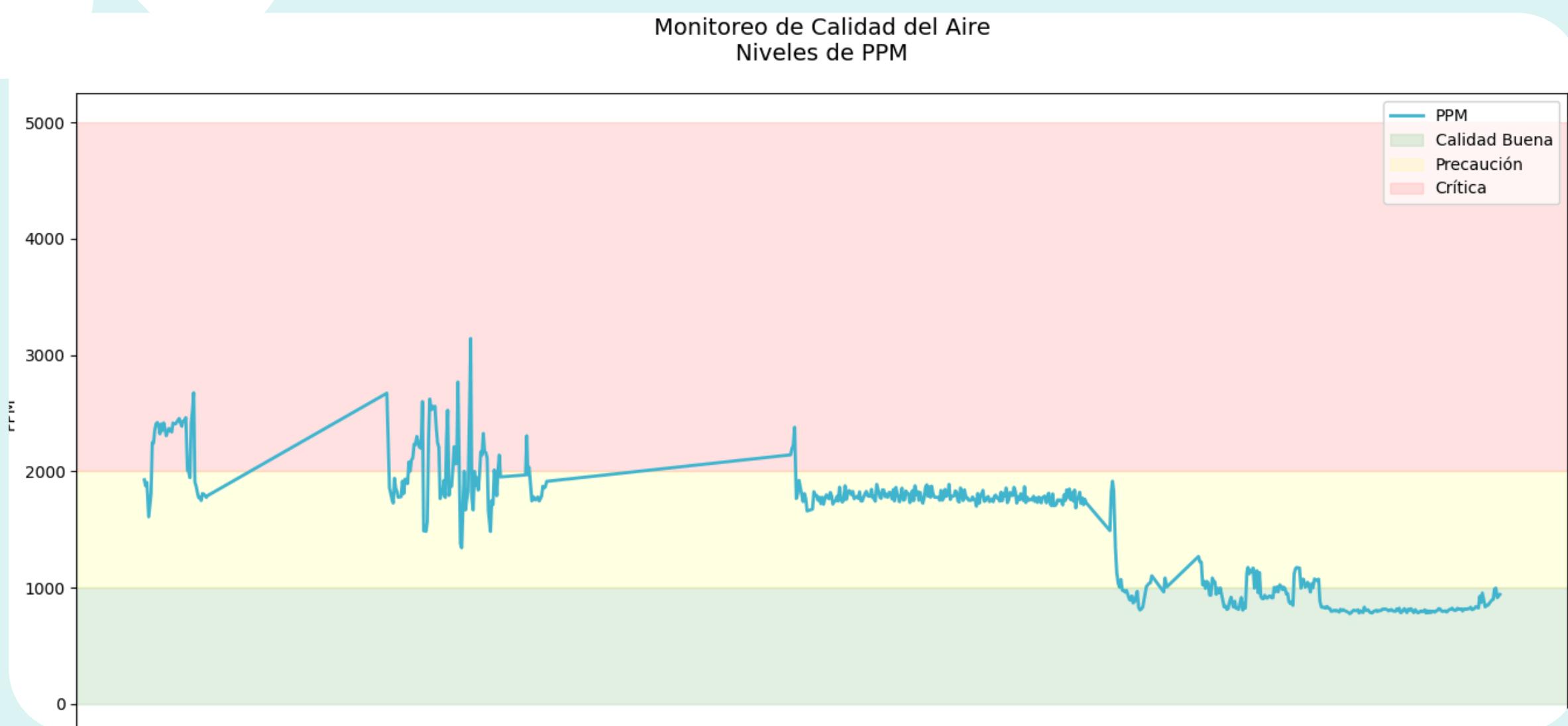
## Preocupaciones:

La temperatura es demasiado alta para un almacén de alimentos, lo que podría acelerar el deterioro de los productos

- La humedad está peligrosamente baja, lo que podría causar:

- \* Deshidratación de productos frescos
- \* Pérdida de calidad en alimentos sensibles
- \* Posible daño a empaques y envases

# CSV - Calidad de Aire y niveles de PPM



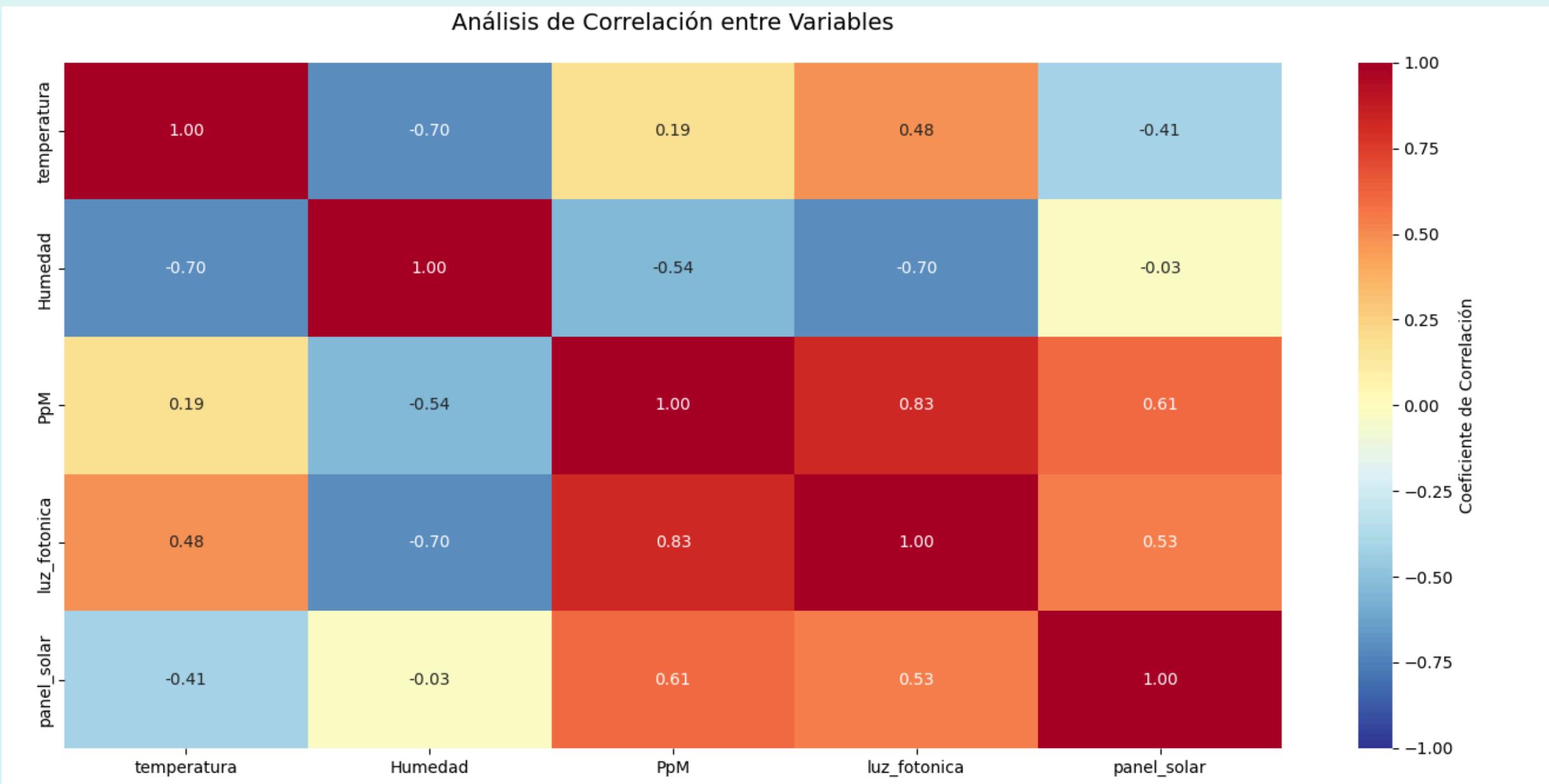
# CSV - Calidad de Aire y niveles de PPM

PPM promedio: 1489

PPM máximo: 3141

PPM mínimo: 775

Lecturas críticas: 75



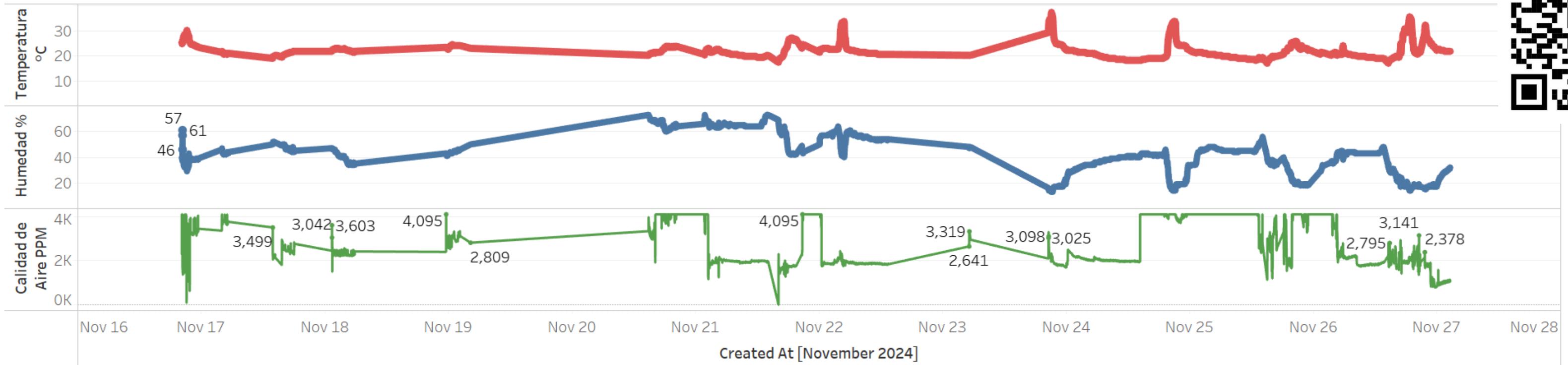
## Observaciones:

- Cuando la temperatura aumenta, la humedad tiende a disminuir significativamente.
- PPM y Panel Solar (0.61): Esta correlación positiva sugiere que los niveles de partículas son mayores durante las horas de sol, cuando el panel solar está más activo.

# Recolección de Datos [Tableau]



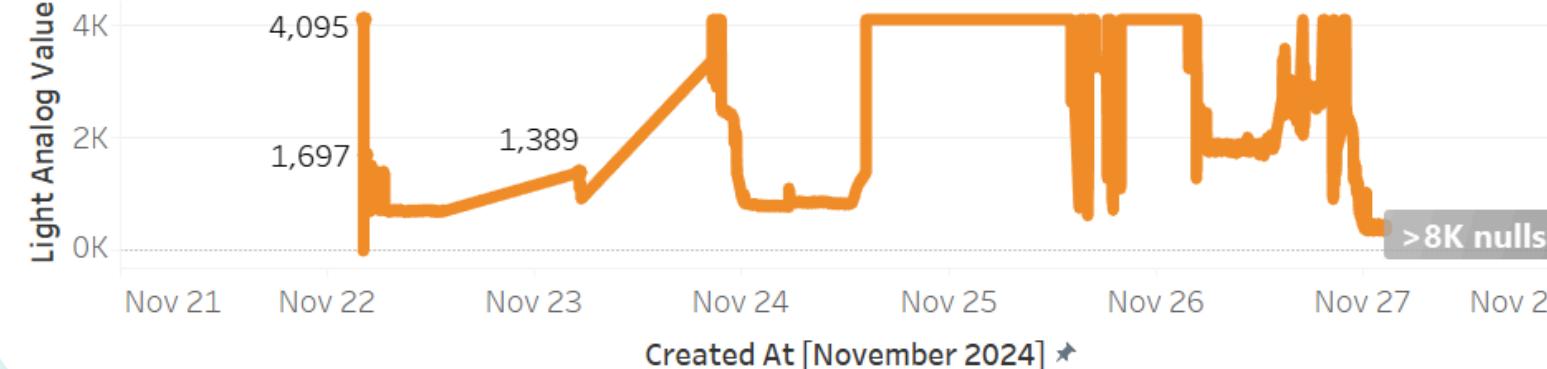
## Ambiente



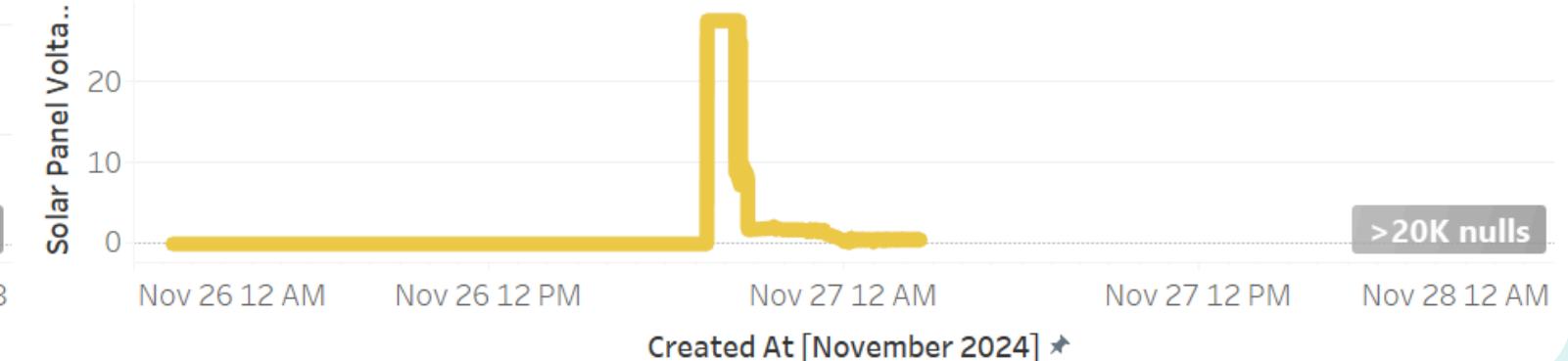
## Distancia

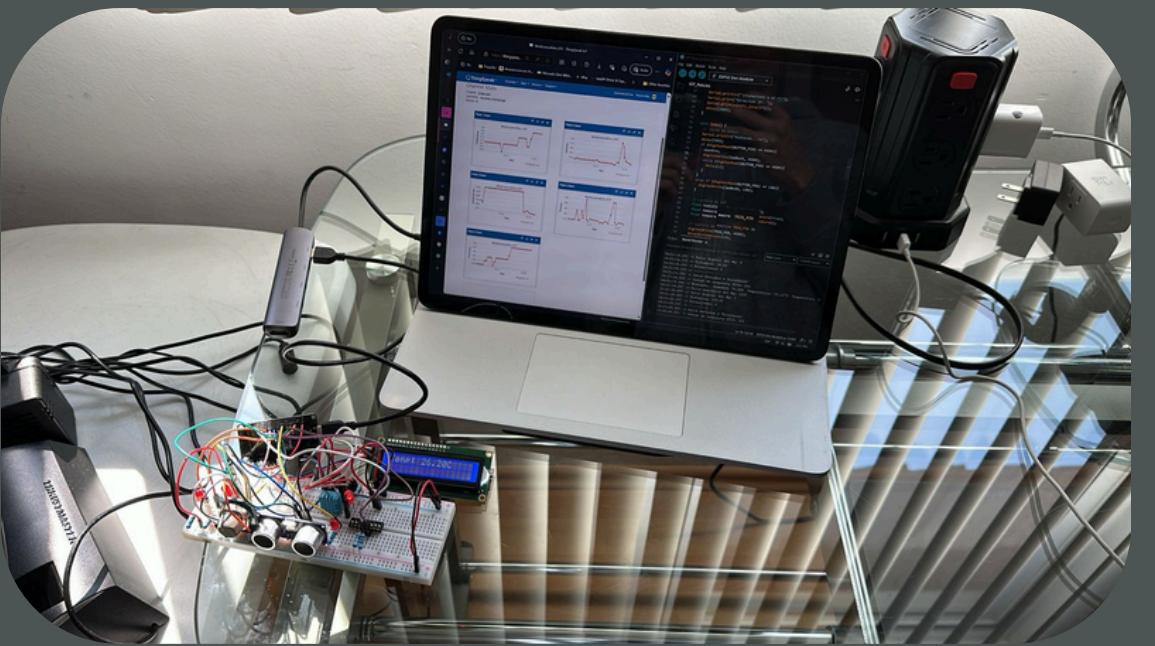


## Luz



## Panel S.





# Conclusión.



## Implementar elementos vistos a lo largo del curso

Además de poner a prueba nuestros conocimientos, aplicamos conceptos y elementos que vimos a lo largo de las clases de todo el módulo, desde el diseño de la interfaz, hasta la programación y conexión de todos los sensores y periféricos para utilizarlos en un caso que hoy en día es muy poco conocido.



# ¡Gracias!

