

# PROYECTO FINAL

## SISTEMA DE SENsoRES PARA DOMÓTICA EN EL HOGAR

Máximo Arenas Roa  
Mary Tere Füguemann Sardá  
Dirk Anton Topcic Martinez  
José Pablo Hernández Alonso

# **DEFINICIÓN DE USUARIO FINAL**

**Persona en el hogar que busque mejorar la comodidad y la eficiencia del hogar mediante la automatización de tareas diarias y la monitorización de diferentes parámetros del entorno y que cuente con acceso a internet**

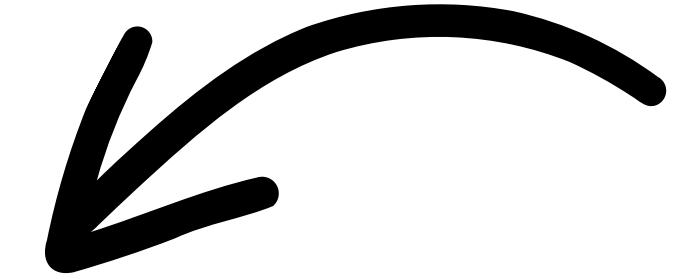
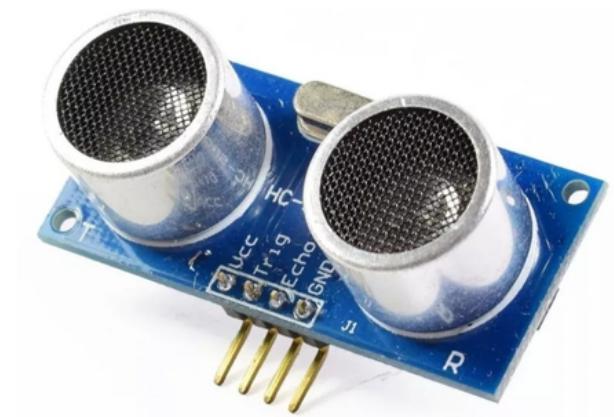
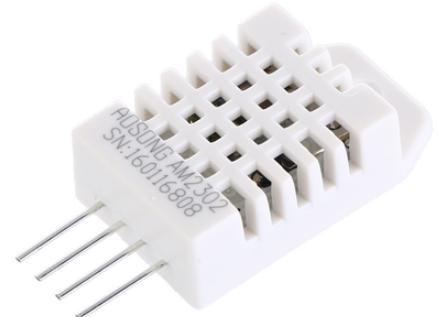


# **INFORMACIÓN QUE NECESITAMOS MEDIR**

- Presencia
- Proximidad
- Humedad
- Tiempo



# **SENsoRES SELECCIONADOS:**



**Sensores de Movimiento (PIR),**

**Sensores de Temperatura y Humedad**

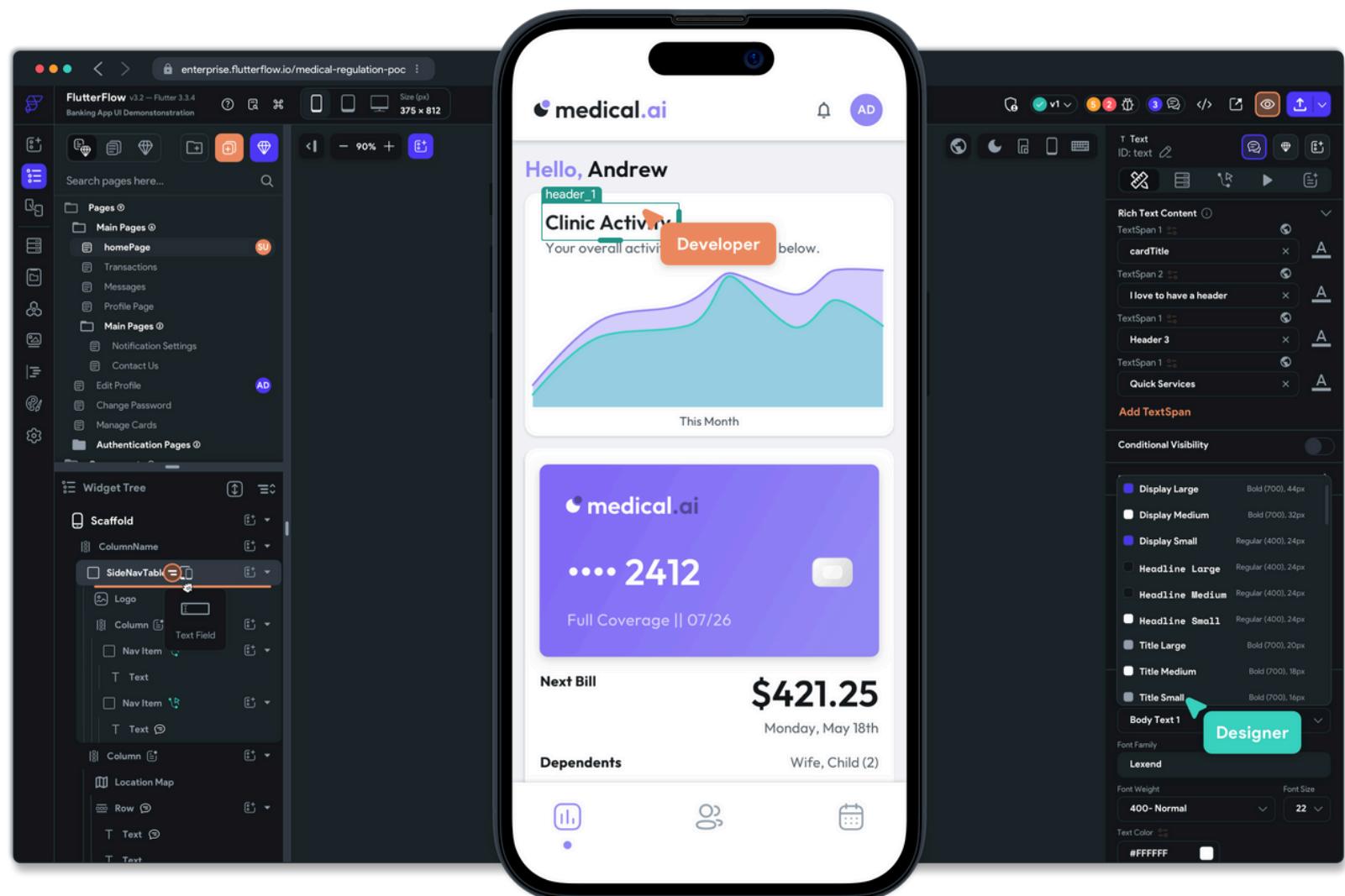
**DHT22, sensor de distancia ultrasónico**

**(HC-SR04), Botones (Temporizador).**

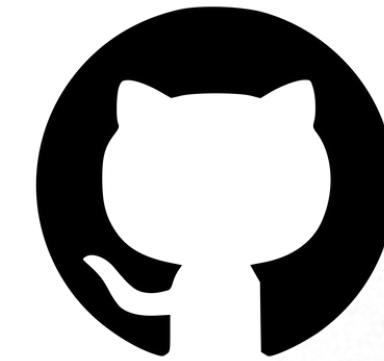
# ARQUITECTURA DE RED



Firebase

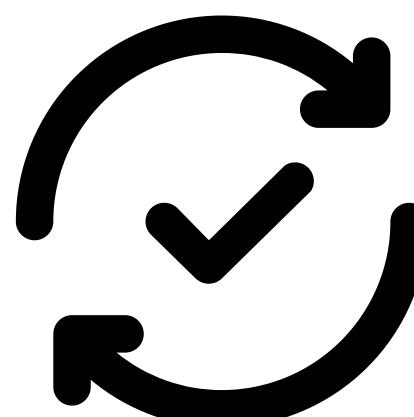
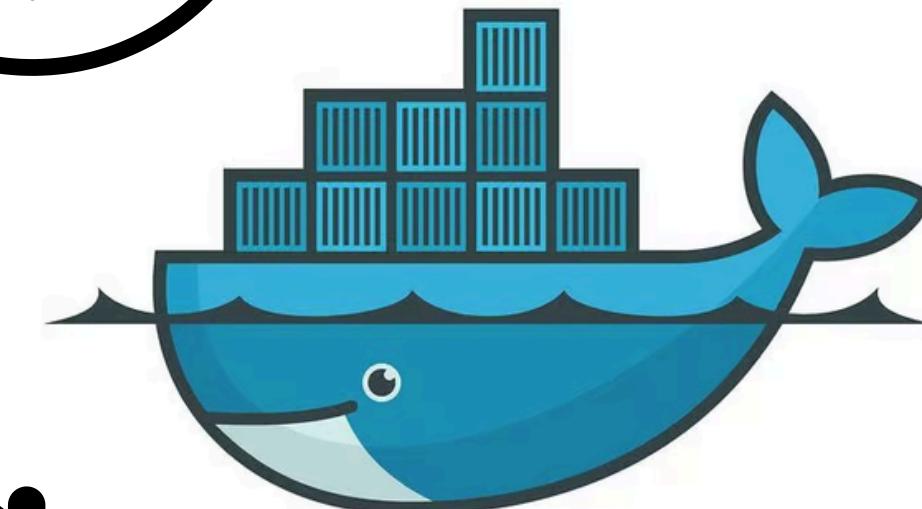
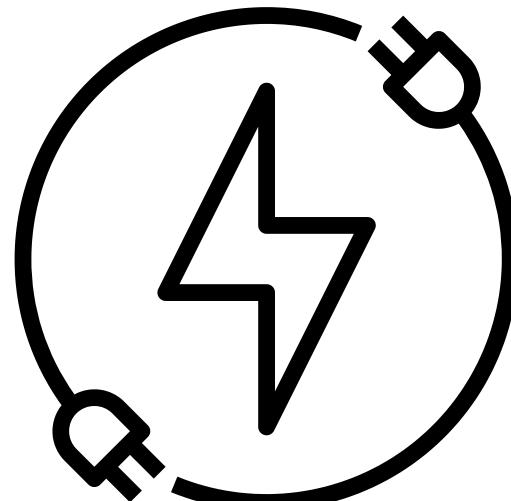


Render



# REQUERIMIENTOS ESPECIALES

- Conexión eléctrica
- WIFI – acceso a internet
- Actualizaciones pausadas
- Manejo de Docker



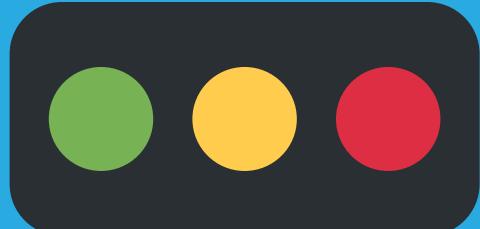
# BOCETO DE DASHBOARD DE INFORMACIÓN

Fecha/ Hora  
Usuario

## Sistema de Domótica

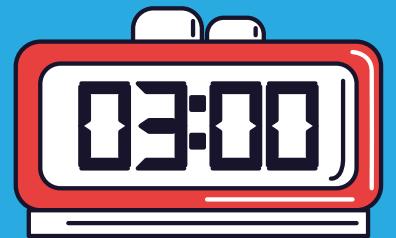
### Cochera

Detector de coche bien estacionado



### Cocina

Temporizador digital



### Baño

Estado de humedad



### Cuarto

Detector de presencia



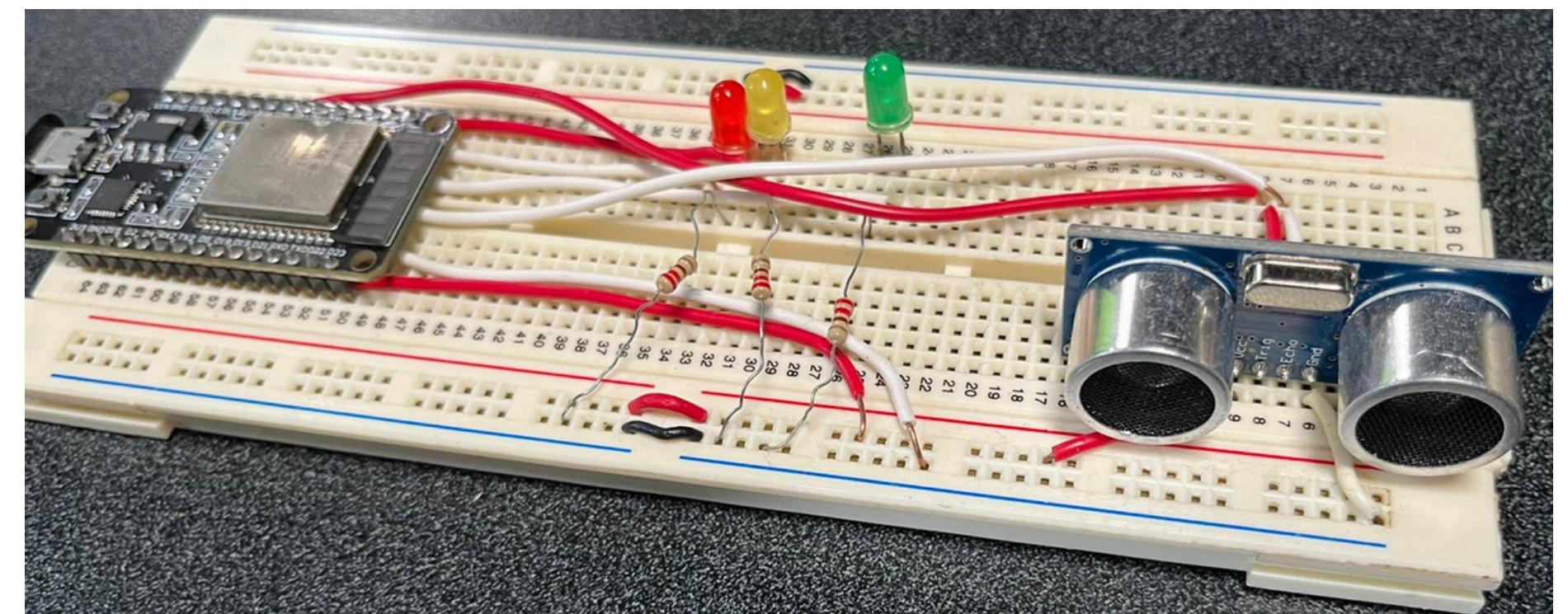
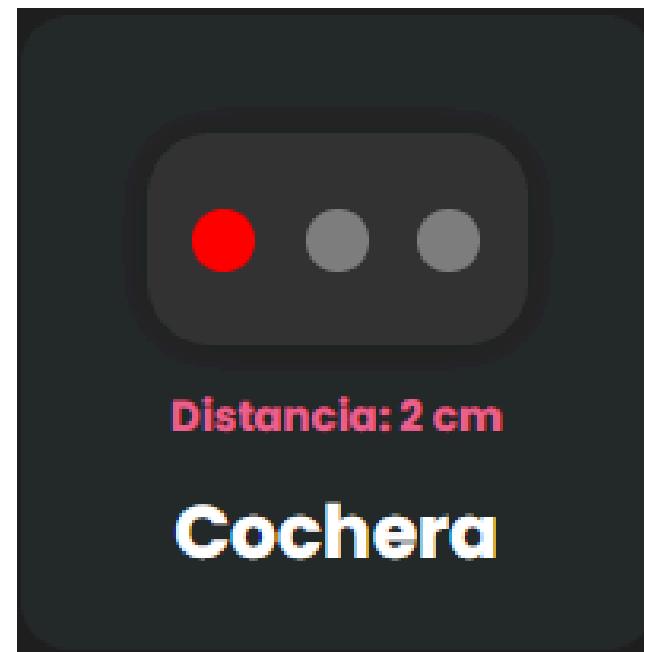
# CONSTRUCCIÓN DE CADA SENSOR

## Cochera

The screenshot displays a development environment for an ESP32-based project. On the left, the code editor shows the `main.c` file with the following content:

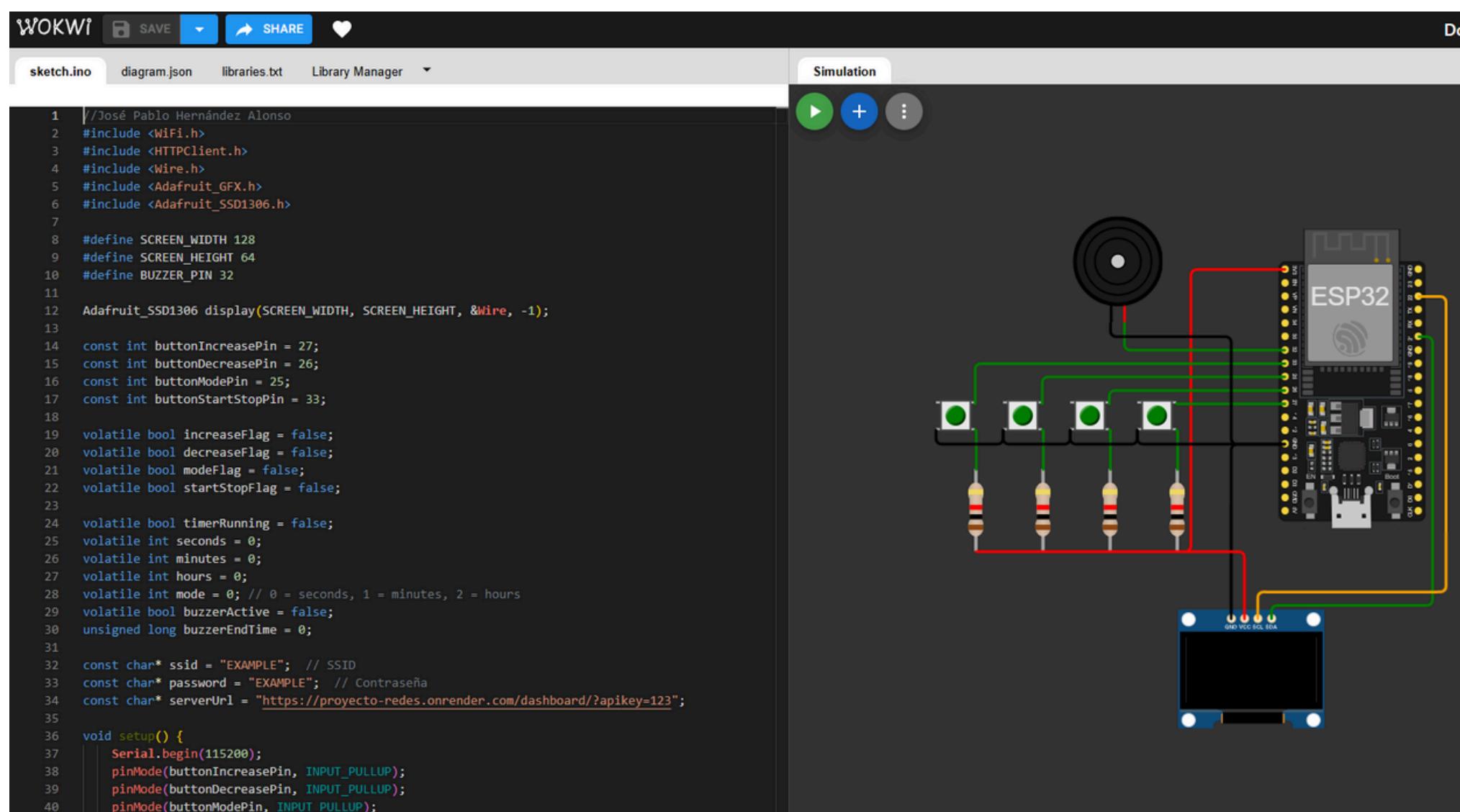
```
1 //Dirk Anton Topic Martinez
2 #include <WiFi.h>
3 #include <HTTPClient.h>
4
5 #define PIN_TRIGGER 26
6 #define PIN_ECHO 27
7
8 int ledV = 32;
9 int ledA = 33;
10 int ledR = 25;
11
12 //PHP server
13 String url = "https://proyecto-redes.onrender.com/dashboard/?";
14 String sensor = "DATO";
15
16
17 void setup(){
18     pinMode(LED_V, OUTPUT);
19     pinMode(LED_A, OUTPUT);
20     pinMode(LED_R, OUTPUT);
21
22     pinMode(PIN_TRIGGER, OUTPUT);
23     pinMode(PIN_ECHO, INPUT);
24
25     delay(1000);
26     Serial.begin(115200);
27
28     WiFi.begin("EXAMPLE", "EXAMPLE");
29
30     while((WiFi.status() != WL_CONNECTED)) {
31         delay(500);
32         Serial.print(".");
33     }
34     Serial.println("Wifi connected");
35 }
36
37 void loop(){
38     delay(1000);
39 }
```

The right side of the interface is a simulation window titled "Simulation". It shows a schematic diagram of the hardware setup. An ESP32 module is connected to three HC-SR04 ultrasonic sensors and three LEDs. The HC-SR04 modules are connected to pins 26 (TRIG) and 27 (ECHO) on the ESP32. The three LEDs are connected to pins 32 (red), 33 (yellow), and 25 (green) on the ESP32. The simulation window includes a toolbar with play, add, and settings buttons.



# CONSTRUCCIÓN DE CADA SENSOR

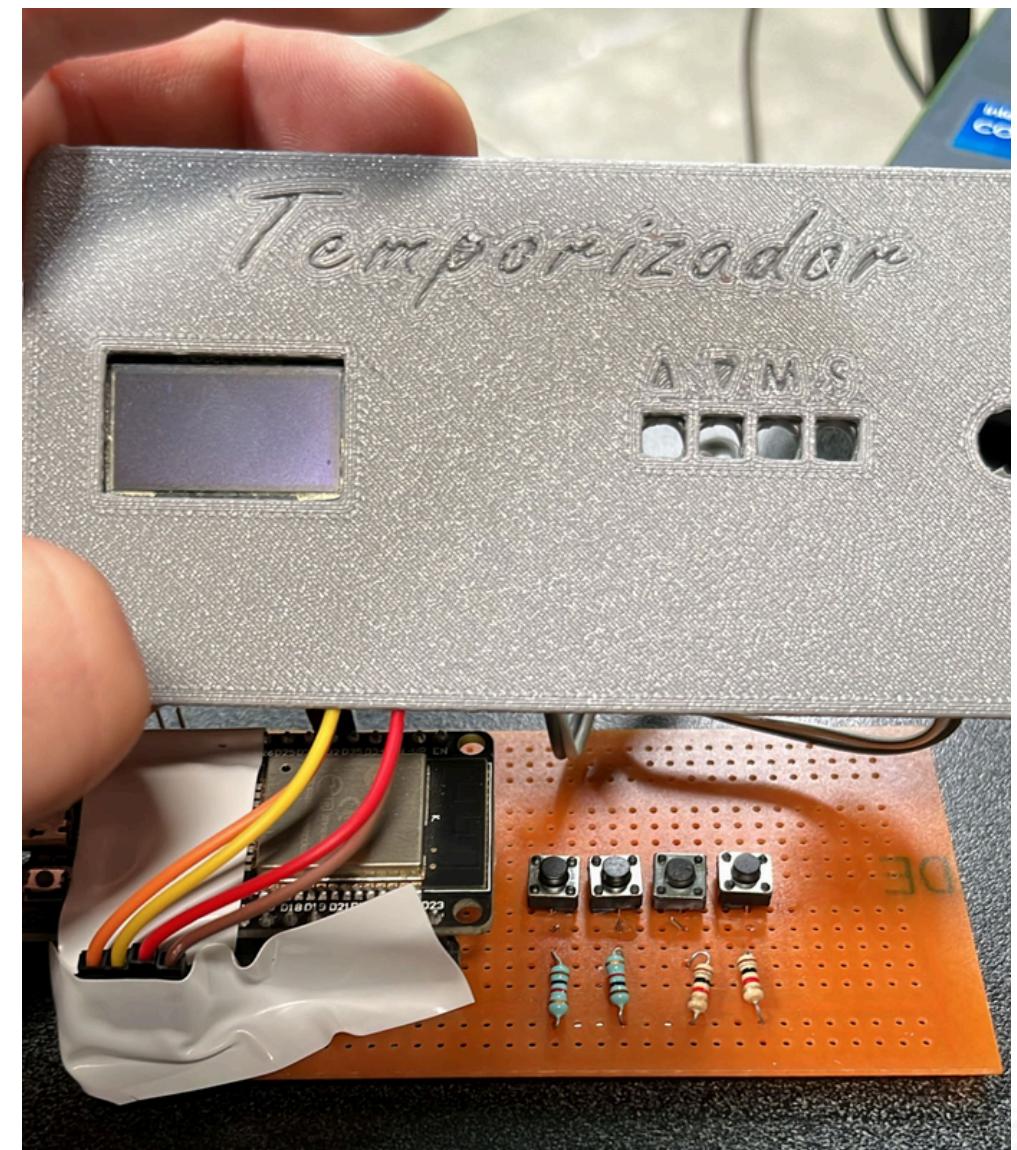
## Cocina



The screenshot shows the Wokwi simulation interface. On the left, the code editor displays the following sketch:

```
1 //José Pablo Hernández Alonso
2 #include <WiFi.h>
3 #include <HTTPClient.h>
4 #include <Wire.h>
5 #include <Adafruit_GFX.h>
6 #include <Adafruit_SSD1306.h>
7
8 #define SCREEN_WIDTH 128
9 #define SCREEN_HEIGHT 64
10#define BUZZER_PIN 32
11
12 Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire, -1);
13
14 const int buttonIncreasePin = 27;
15 const int buttonDecreasePin = 26;
16 const int buttonModePin = 25;
17 const int buttonStartStopPin = 33;
18
19 volatile bool increaseFlag = false;
20 volatile bool decreaseFlag = false;
21 volatile bool modeFlag = false;
22 volatile bool startStopFlag = false;
23
24 volatile bool timerRunning = false;
25 volatile int seconds = 0;
26 volatile int minutes = 0;
27 volatile int hours = 0;
28 volatile int mode = 0; // 0 = seconds, 1 = minutes, 2 = hours
29 volatile bool buzzerActive = false;
30 unsigned long buzzerEndTime = 0;
31
32 const char* ssid = "EXAMPLE"; // SSID
33 const char* password = "EXAMPLE"; // Contraseña
34 const char* serverUrl = "https://proyecto-redes.onrender.com/dashboard?apikey=123";
35
36 void setup() {
37     Serial.begin(115200);
38     pinMode(buttonIncreasePin, INPUT_PULLUP);
39     pinMode(buttonDecreasePin, INPUT_PULLUP);
40     pinMode(buttonModePin, INPUT_PULLUP);
```

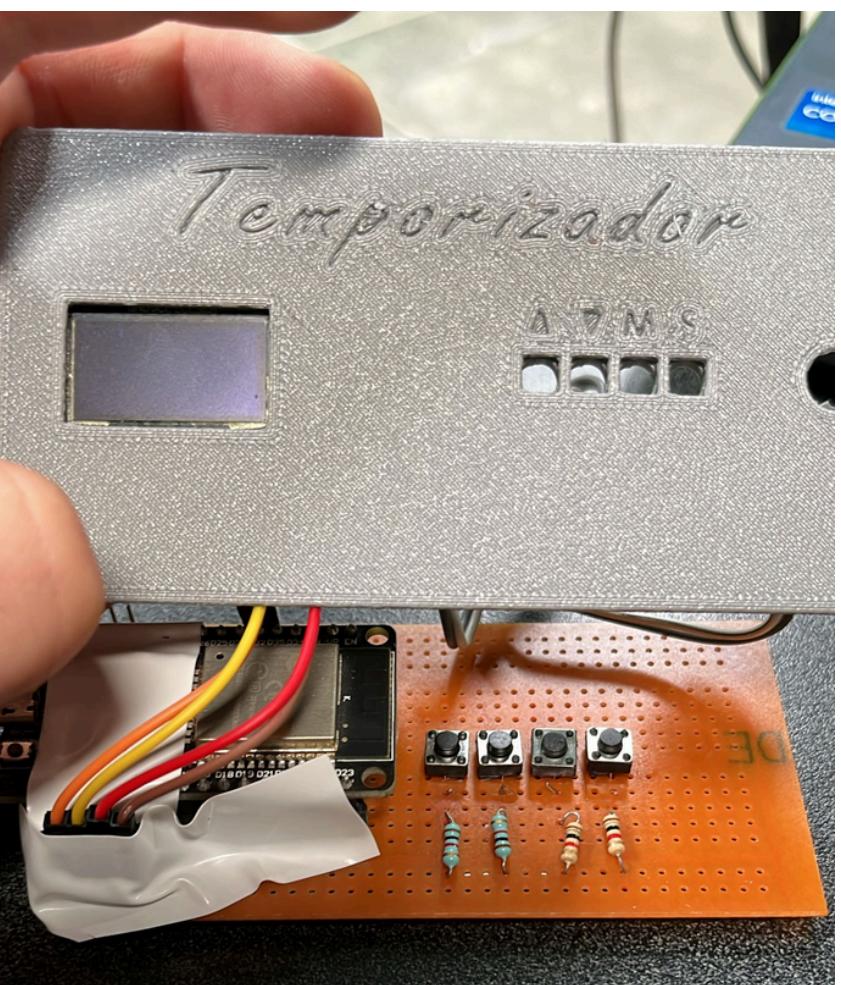
The circuit diagram on the right shows an ESP32 microcontroller connected to a 0.96" SSD1306 display, four push buttons, and a buzzer. The display is connected to pins D4, D5, and D6. The push buttons are connected to pins 27, 26, 25, and 33. The buzzer is connected to pin 32.



00:00:00

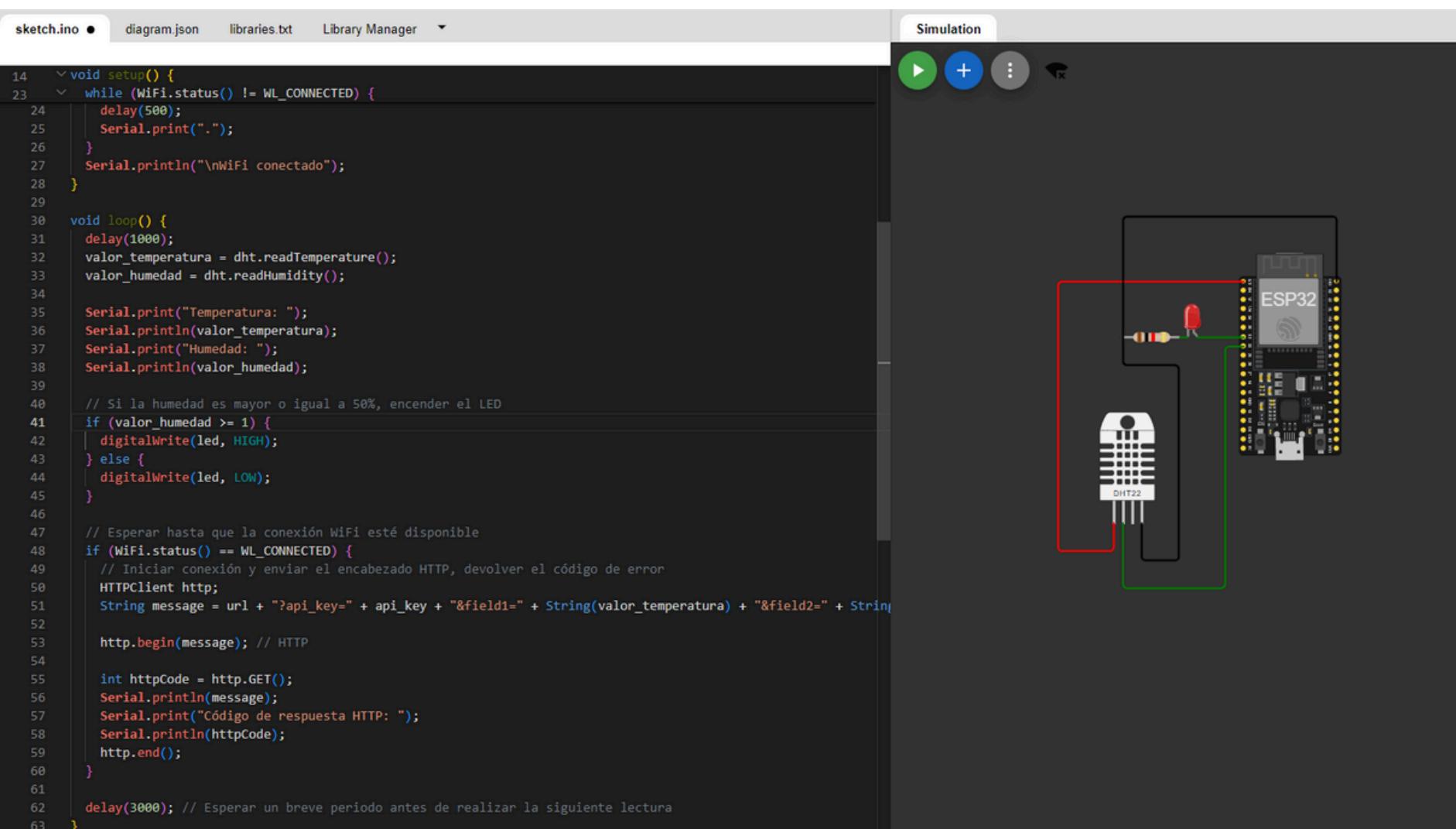
Status: terminado

Cocina



# CONSTRUCCIÓN DE CADA SENSOR

## Baño



```
sketch.ino • diagram.json libraries.txt Library Manager ▾

14 void setup() {
23   while (WiFi.status() != WL_CONNECTED) {
24     delay(500);
25     Serial.print(".");
26   }
27   Serial.println("\nWiFi conectado");
28 }

29 void loop() {
30   delay(1000);
31   valor_temperatura = dht.readTemperature();
32   valor_humedad = dht.readHumidity();
33
34   Serial.print("Temperatura: ");
35   Serial.println(valor_temperatura);
36   Serial.print("Humedad: ");
37   Serial.println(valor_humedad);
38
39   // Si la humedad es mayor o igual a 50%, encender el LED
40   if (valor_humedad >= 50) {
41     digitalWrite(led, HIGH);
42   } else {
43     digitalWrite(led, LOW);
44   }
45
46
47   // Esperar hasta que la conexión WiFi esté disponible
48   if (WiFi.status() == WL_CONNECTED) {
49     // Iniciar conexión y enviar el encabezado HTTP, devolver el código de error
50     HttpClient http;
51     String message = url + "?api_key=" + api_key + "&field1=" + String(valor_temperatura) + "&field2=" + String(valor_humedad);
52
53     http.begin(message); // HTTP
54
55     int httpCode = http.GET();
56     Serial.println(message);
57     Serial.print("Código de respuesta HTTP: ");
58     Serial.println(httpCode);
59     http.end();
60   }
61
62   delay(3000); // Esperar un breve periodo antes de realizar la siguiente lectura
63 }
```

Simulation

```
$TEXTO = $DATO_var . "\r\n";
// Guarda el valor en el archivo 'datos.txt'
file_put_contents("datos.txt", $TEXTO);

// Lee el contenido del archivo 'datos.txt'
$IVO = file_get_contents("datos.txt");
$i = strpos($IVO, "\r\n");
// Extrae el valor de humedad del archivo
$lectura = substr($IVO, 0, $pos1);

TYPE html
lang="es"
-
meta charset="UTF-8"
meta name="viewport" content="width=device-width, initial-scale=1.0"
title>Sensor de Humedad</title>
style
body {
  font-family: Arial, sans-serif;
  display: flex;
  justify-content: center;
  align-items: center;
  height: 100vh;
  margin: 0;
  background-color: #000;
  color: #fff;
}
.sensor-container {
  background: #333;
  padding: 20px;
  border-radius: 10px;
  box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
  text-align: center;
  width: 100px;
}
.thermometer {
  width: 50px;
  height: 300px;
  border: 2px solid #fff;
  border-radius: 25px;
  position: relative;
  background: #e0e0e0;
}
.thermometer-fill {
  width: 100%;
  position: absolute;
  bottom: 0;
  border-radius: 0 0 25px 25px;
}

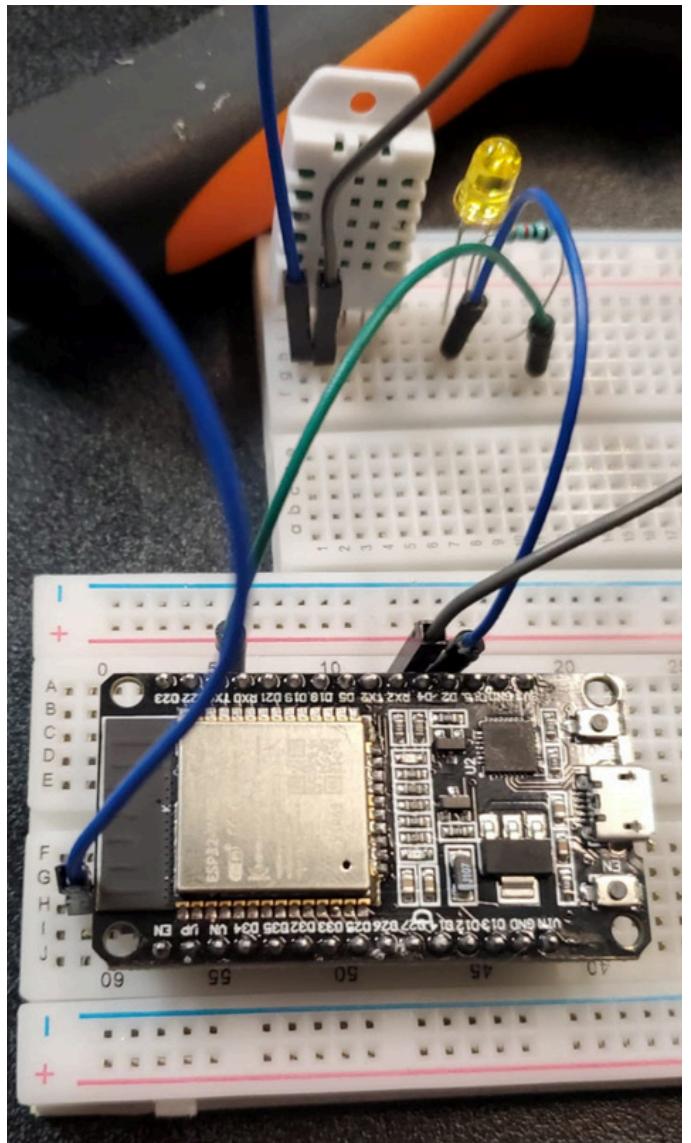
Sensor de Humedad
10%
80%
60%
40%
20%
0%
Humedad: 70%
```

Console

```
[Mon Jul 8 17:51:09 2024] PHP 8.2.0RC7 Development Server (http://0.0.0.0:8000)
[Mon Jul 8 17:51:11 2024] 172.31.196.61:60648 Accepted
[Mon Jul 8 17:51:11 2024] 172.31.196.61:60648 [200]: GET /
[Mon Jul 8 17:51:11 2024] 172.31.196.61:60648 Closing
[Mon Jul 8 17:51:16 2024] 172.31.196.61:39042 Accepted
[Mon Jul 8 17:51:16 2024] 172.31.196.61:39042 [200]: GET /
[Mon Jul 8 17:51:16 2024] 172.31.196.61:39042 Closing
[Mon Jul 8 17:51:16 2024] 172.31.196.61:39050 Accepted
[Mon Jul 8 17:51:16 2024] 172.31.196.61:39050 [200]: GET /
[Mon Jul 8 17:51:16 2024] 172.31.196.61:39050 Closing
[Mon Jul 8 17:51:16 2024] 172.31.196.61:39050 Accepted
[Mon Jul 8 17:51:16 2024] 172.31.196.61:39050 [200]: GET /
[Mon Jul 8 17:51:16 2024] 172.31.196.61:39050 Closing
```

Run

Port:8000 opened on f...replit.dev



# CONSTRUCCIÓN DE CADA SENSOR

# Cuarto

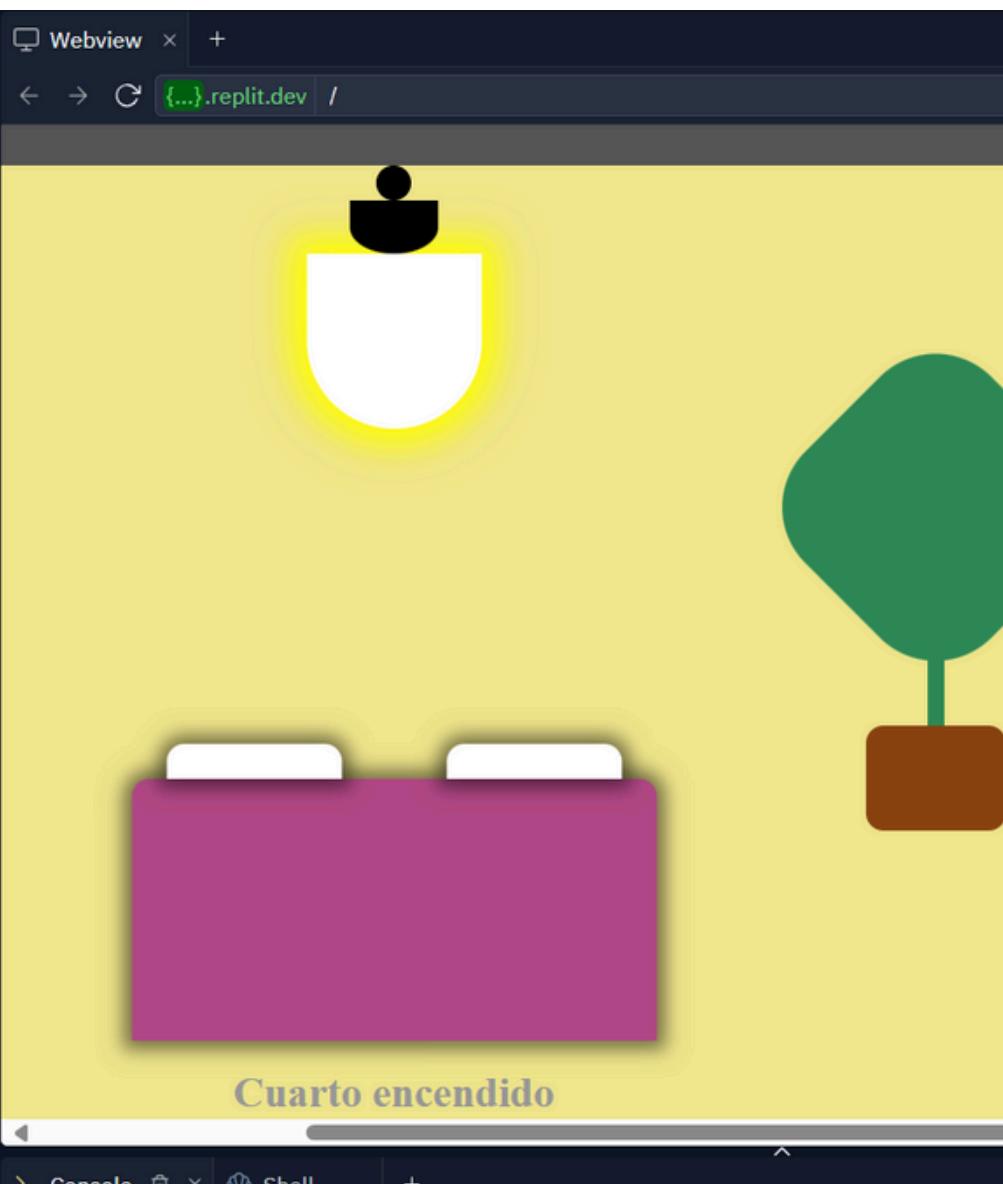
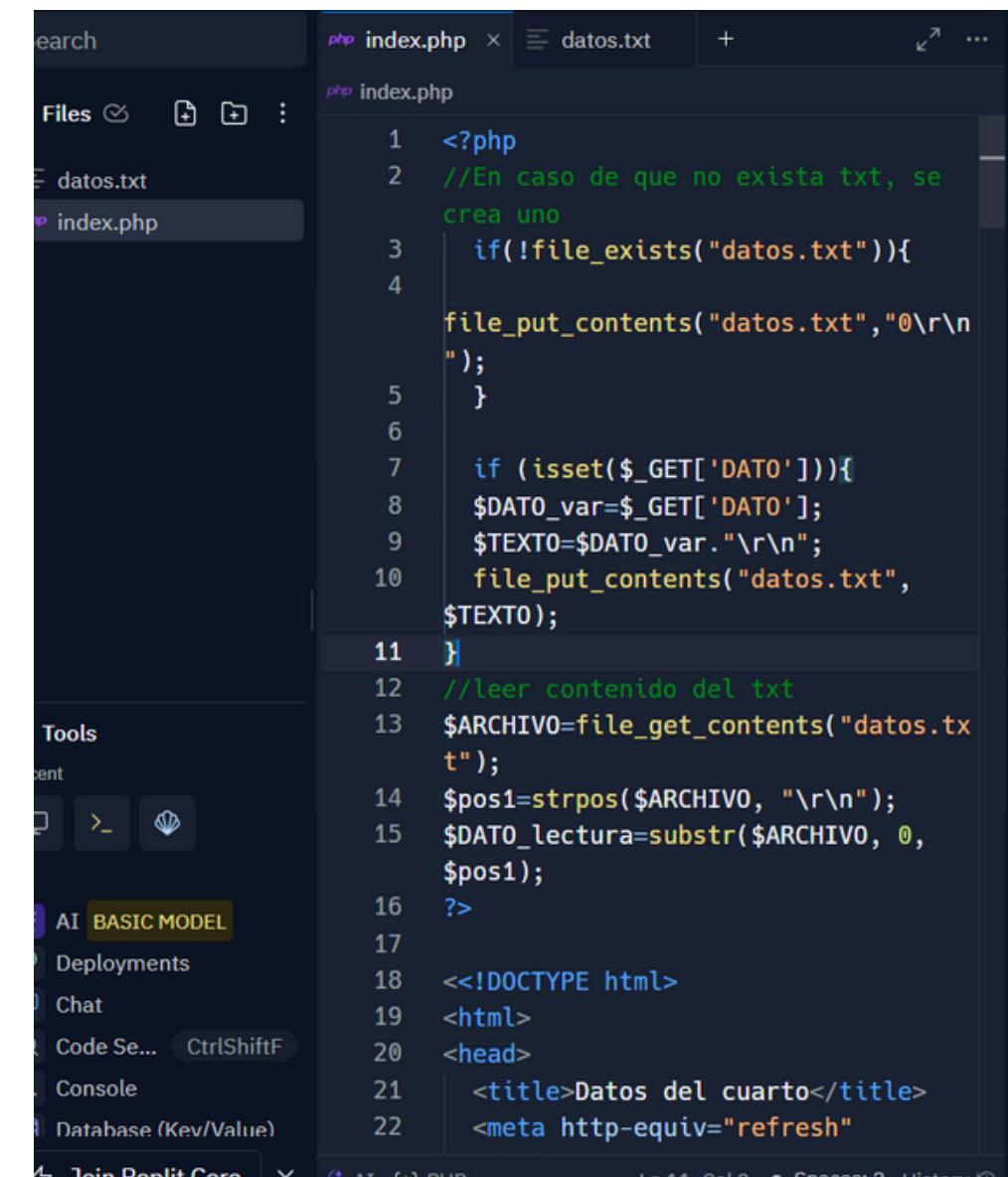
WOKWi PIR proyecto final

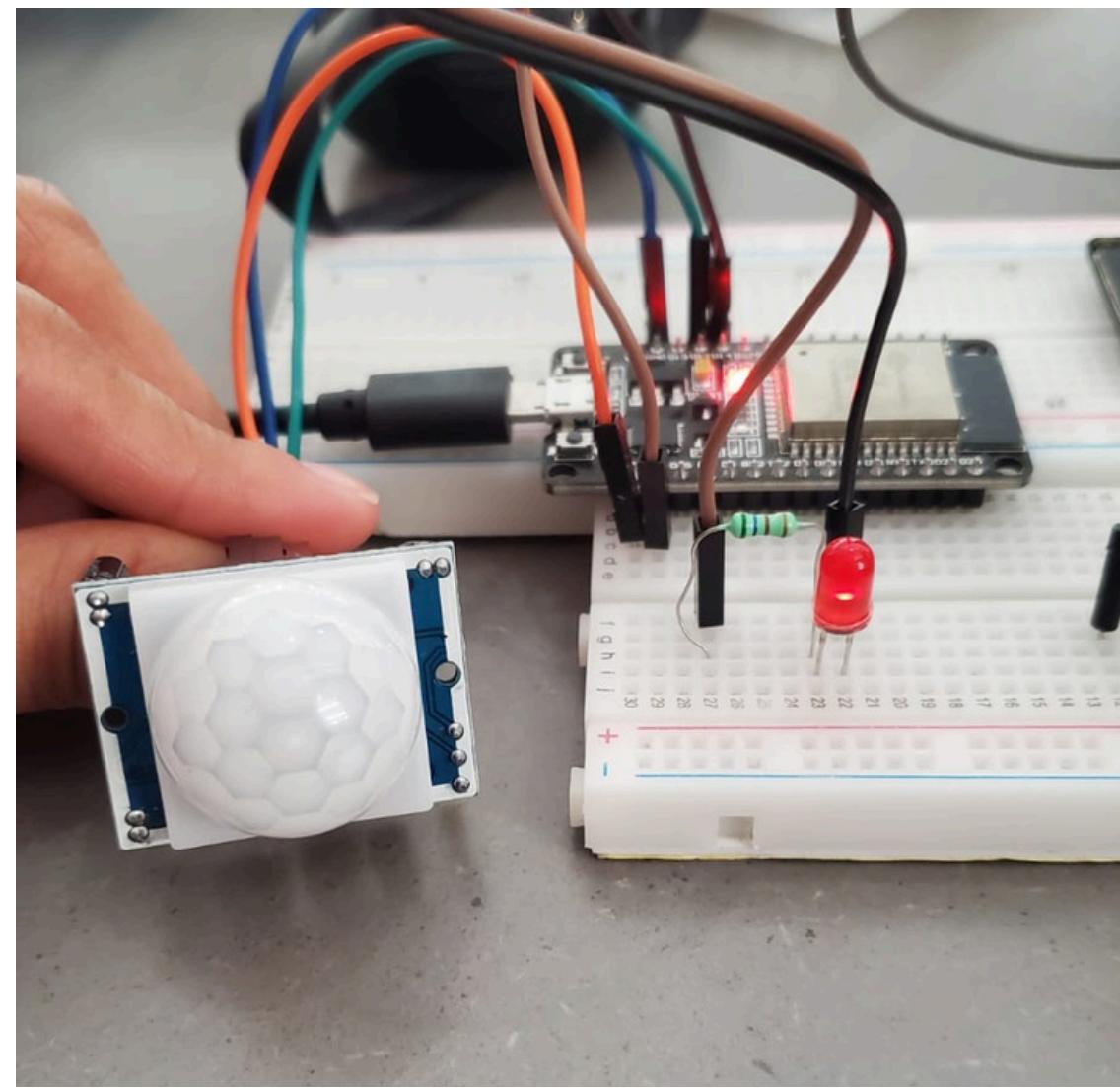
sketch.ino diagram.json Library Manager

```
1 //Envío de datos a Thinkspeak
2 #include <WiFi.h>
3 #include <HTTPClient.h>
4
5 int pir = 32;
6 int led = 33;
7
8 int status_pir = 0;
9
10 String url= "https://api.thingspeak.com/update?";
11 String api_key= "9DBSUNUPPCD8OPRY";
12 String sensor= "field4";
13
14 void setup(){
15   pinMode(led, OUTPUT);
16   pinMode(pir, INPUT);
17
18   delay(1000);
19   Serial.begin(115200);
20
21   WiFi.begin("Wokwi-GUEST", "");
22
23   while((WiFi.status() != WL_CONNECTED) {
24     delay(500);
25     Serial.print(".");
26   }
27   Serial.println("Wifi conected");
28 }
29
30 void loop(){
31   status_pir = digitalRead(pir);
32
33   if (status_pir == 1){
34     digitalWrite(led, HIGH);
```

Simulation

ESP32





# DASHBOARD FINAL DE INFORMACIÓN

Inicio de Sesión

usuario

contraseña

INGRESAR

Crear Cuenta →

Registrar Usuario

usuario

contraseña

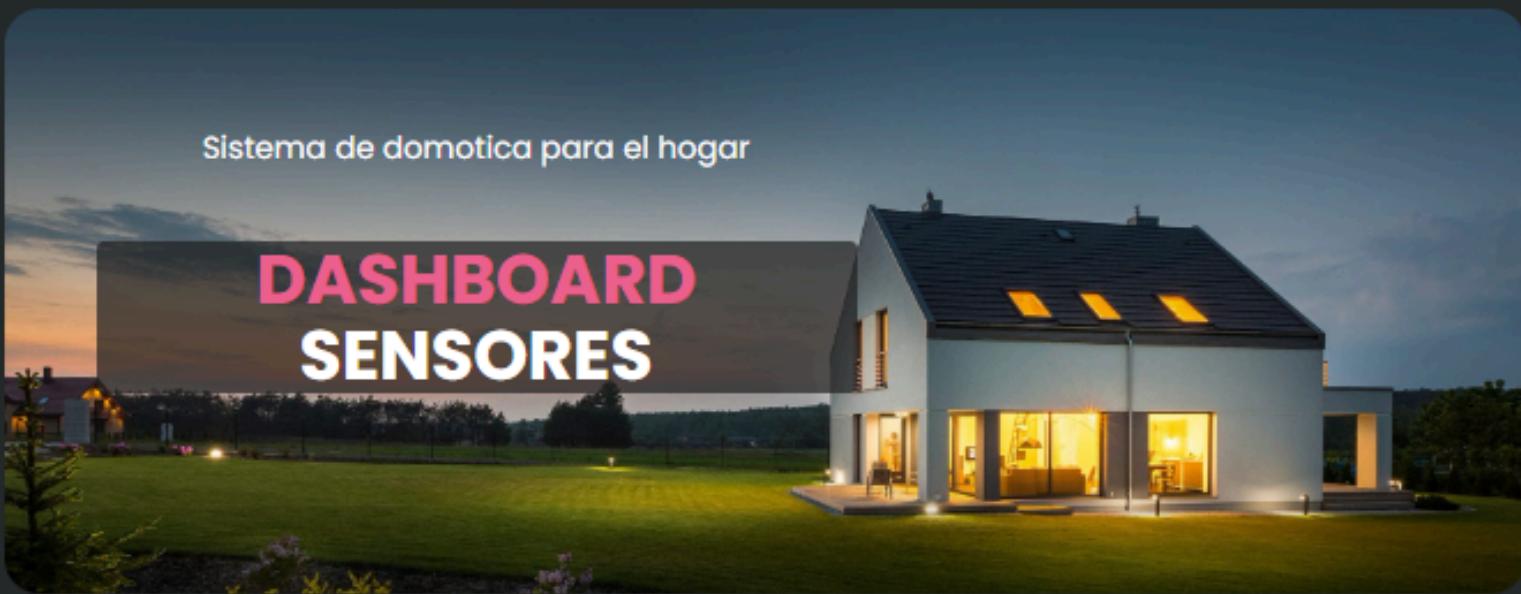
contraseña

CREAR

Regresar →

Sistema de domotica para el hogar

## DASHBOARD SENSORES



### Menu de Sensores

00:00:00

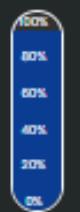
Status: terminado

Cocina



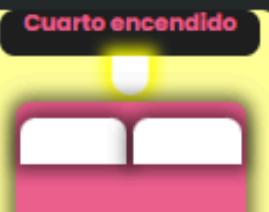
Distancia: 2 cm

Cochera



Humedad: 90%

Baño



Cuarto

# Dashboard



Timer  
[time]  
Terminado



Humedad  
[1]%

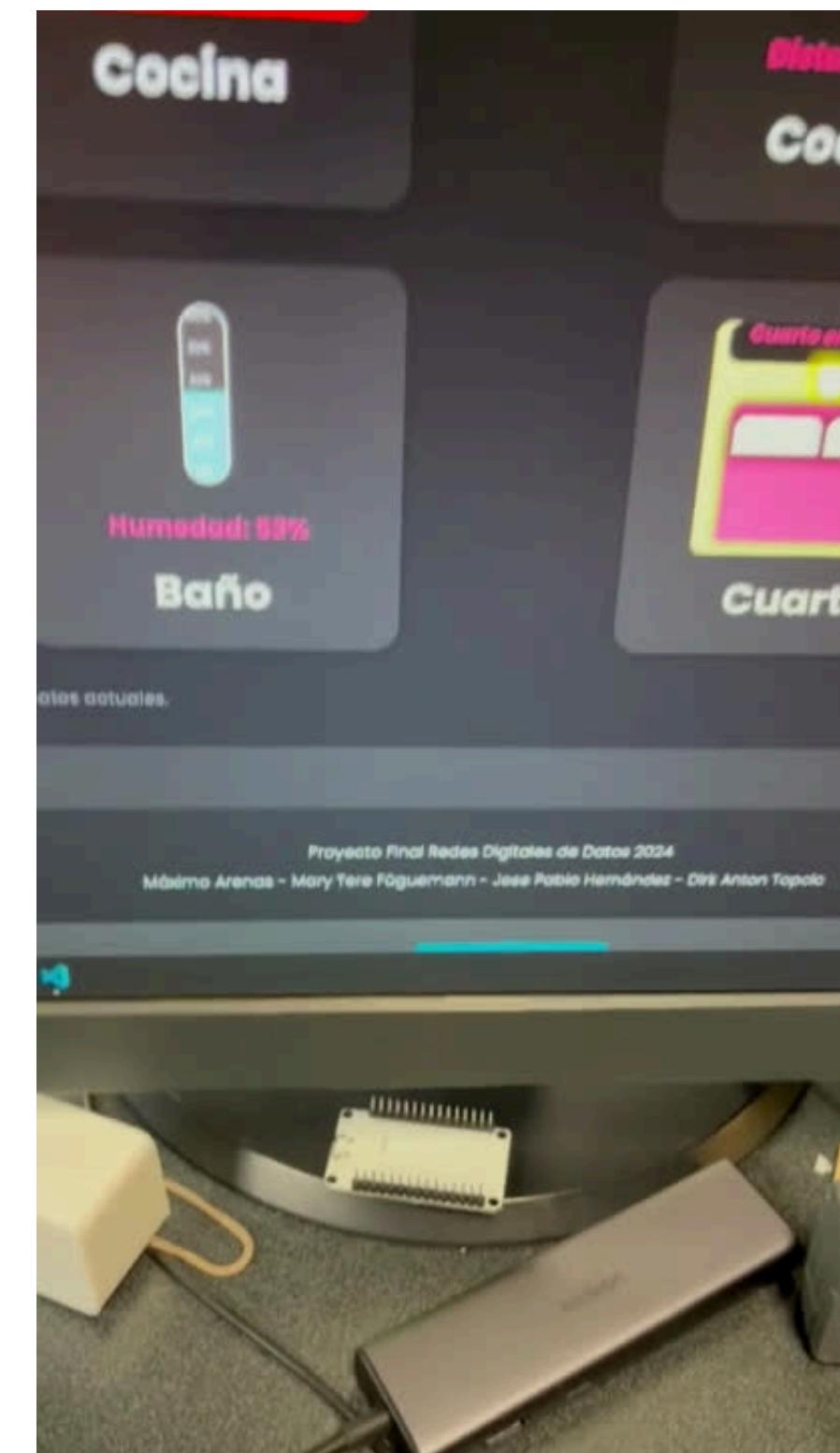
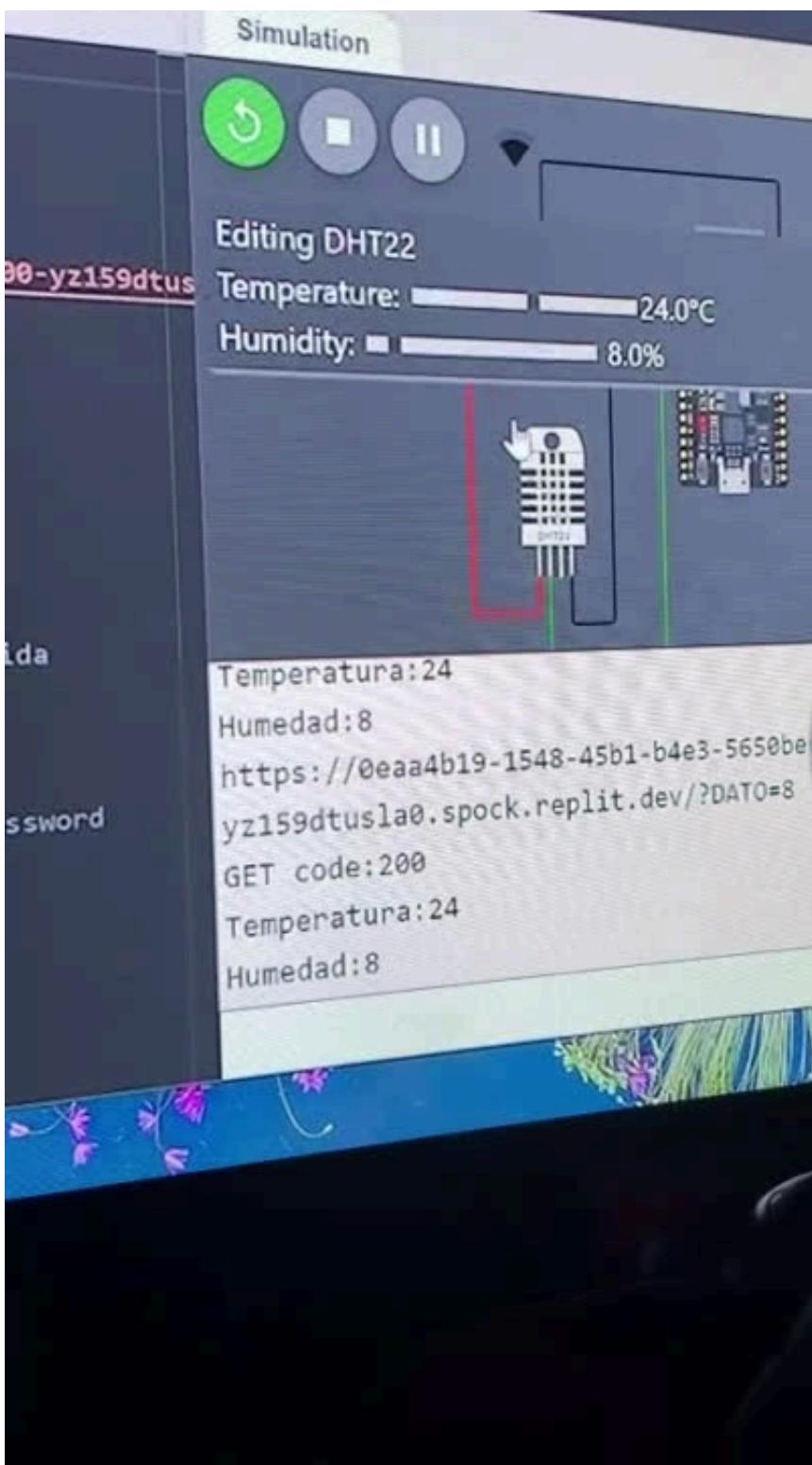


Proximity  
[1]cm

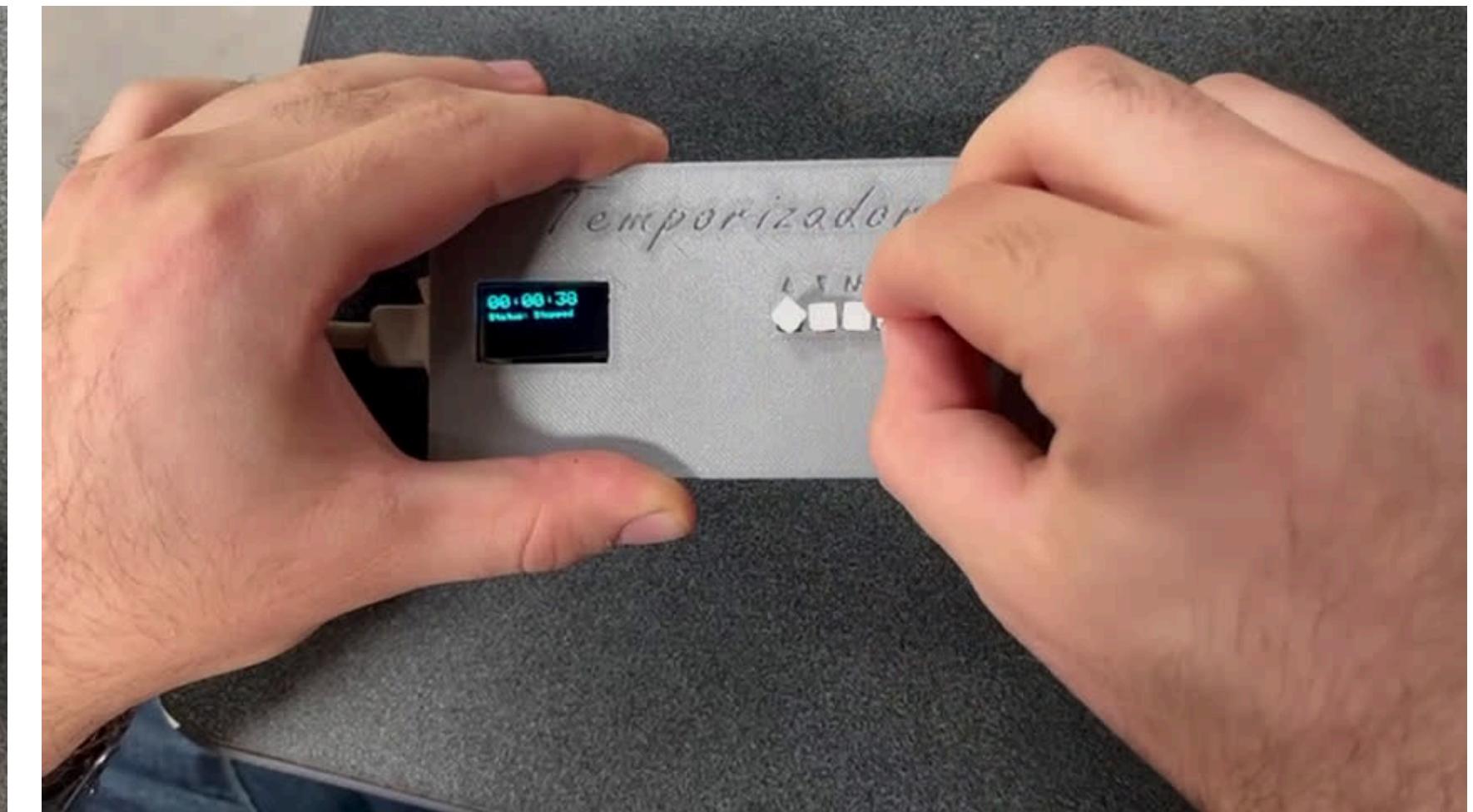


Presencia  
[mov]

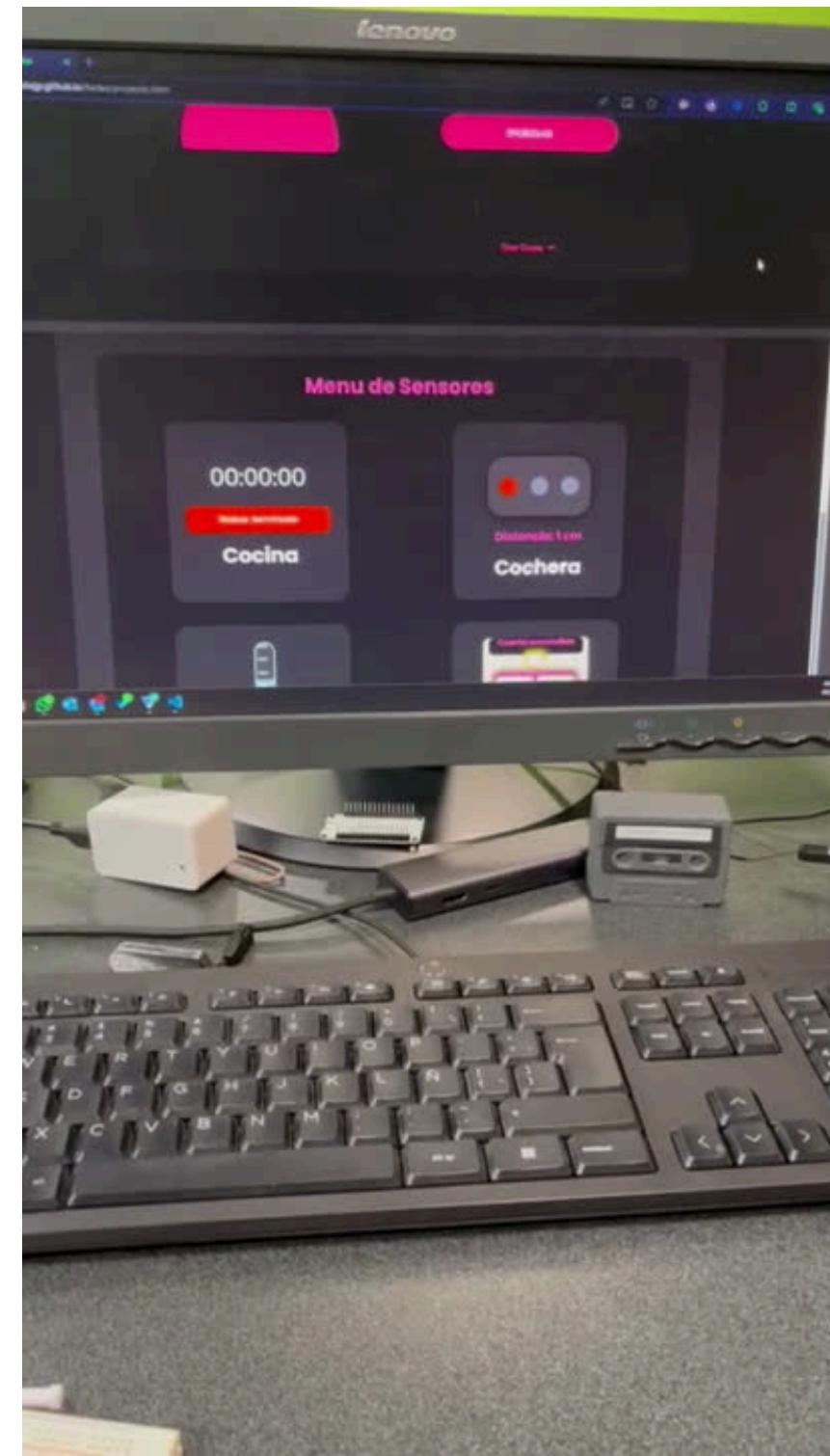
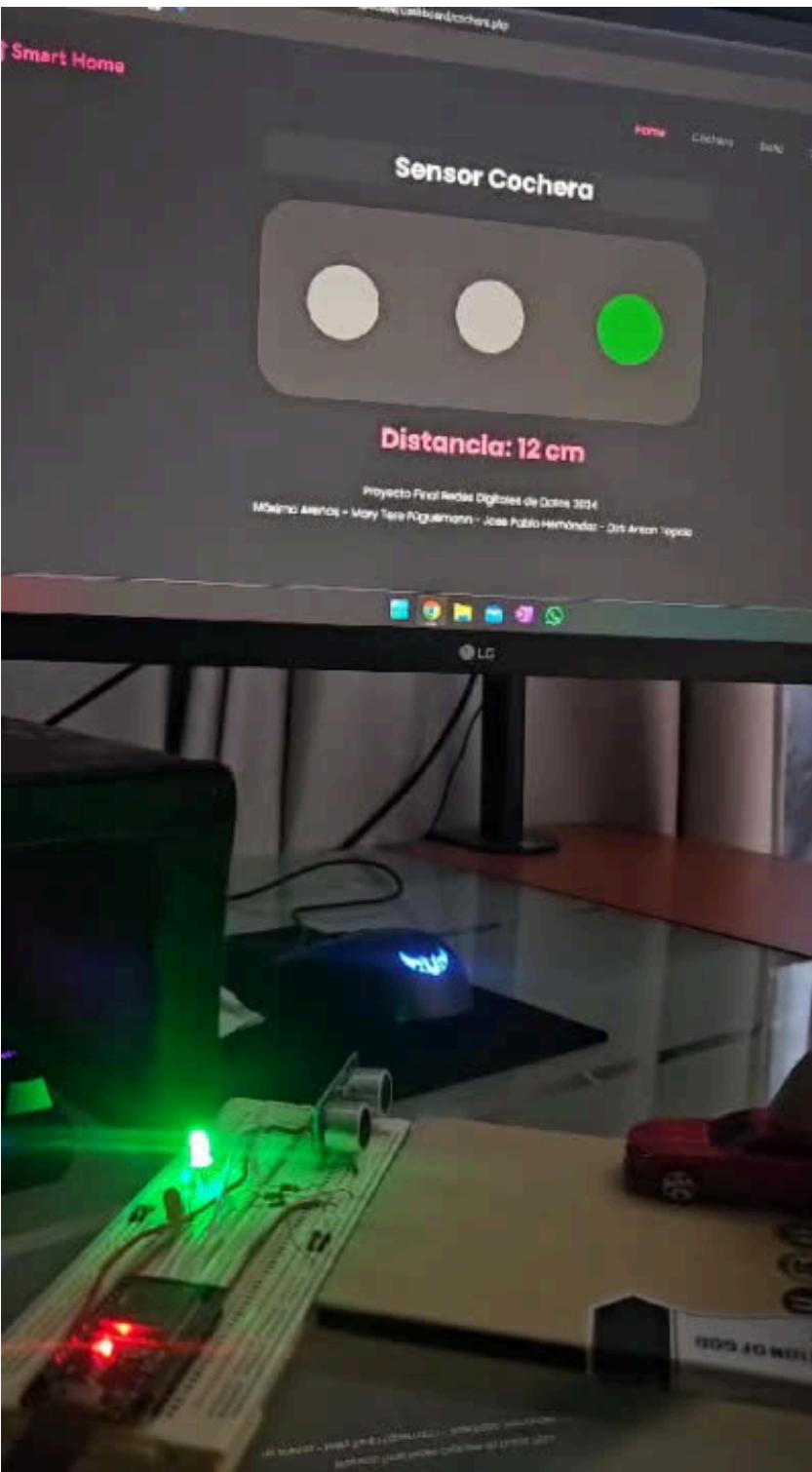
# VIDEOS DE SU FUNCIONAMIENTO



# VIDEOS DE SU FUNCIONAMIENTO



# VIDEOS DE SU FUNCIONAMIENTO



# VIDEOS DE SU FUNCIONAMIENTO

