

SISTEMA DE ALERTA TEMPRANA DE INUNDACIONES

Universidad Distrital Francisco José de Caldas

Por:

Nicolás Méndez Romero 20191573068

Harold Alfonso Velásquez Arias 20191573128

Industria 4.0

2023

1. Resumen

El proyecto propone el diseño e implementación de un sistema de alerta temprana de inundaciones basado en la Industria 4.0. Este sistema estará compuesto por una red de sensores de nivel de agua y estaciones meteorológicas, un sistema de adquisición y procesamiento de datos, un sistema de predicción y alerta temprana de inundaciones, y un sistema de comunicación y difusión de alertas.

El objetivo principal del proyecto es reducir el impacto de las inundaciones en las comunidades y empresas afectadas al proporcionar alertas tempranas precisas y confiables que permitan tomar medidas preventivas con anticipación. Se considerarán aspectos como la selección de sensores adecuados, la implementación de algoritmos de procesamiento y análisis de datos, y la integración de tecnologías de comunicación y difusión de alertas.

El proyecto se enfocará en evaluar la efectividad del sistema en términos de precisión y tiempo de respuesta. Se espera que la implementación de este sistema contribuya a la prevención de inundaciones y protección de vidas y propiedades.

Abstract: The project proposes the design and implementation of a flood early warning system based on Industry 4.0. This system will be composed of a network of water level sensors and weather stations, a data acquisition and processing system, a flood forecasting and early warning system, and a communication and alert dissemination system.

The main objective of the project is to reduce the impact of floods on affected communities and businesses by providing accurate and reliable early warnings that will enable preventive measures to be taken in advance. Aspects such as the selection of appropriate sensors, the implementation of data processing and analysis algorithms, and the integration of communication and warning dissemination technologies will be considered.

The project will focus on evaluating the effectiveness of the system in terms of accuracy and response time. The implementation of this system is expected to contribute to flood prevention and protection of lives and property.

2. Introducción

Las inundaciones son uno de los desastres naturales más frecuentes y devastadores que afectan a muchas regiones del mundo. Cada año, millones de personas se ven afectadas por inundaciones, que causan la pérdida de vidas, propiedades y medios de subsistencia. Los cambios en el clima, la urbanización y el aumento del nivel del mar son algunas de las causas que han aumentado la frecuencia y la intensidad de las inundaciones en todo el mundo. Ante esta situación, es necesario contar con sistemas de alerta temprana de inundaciones que permitan tomar medidas preventivas y minimizar el impacto de este tipo de eventos.

La industria 4.0 ha permitido el desarrollo de tecnologías avanzadas para la recopilación, procesamiento y análisis de datos, lo que ha llevado a la creación de sistemas de alerta temprana más precisos y eficaces. En este anteproyecto se propone el diseño e implementación de un sistema de alerta temprana de inundaciones basado en la industria 4.0.

El sistema estará compuesto por una red de sensores de nivel de agua y estaciones meteorológicas, un sistema de adquisición y procesamiento de datos, un sistema de predicción y alerta temprana de inundaciones, y un sistema de comunicación y difusión de alertas. El objetivo del proyecto es reducir el impacto de las inundaciones en las comunidades y empresas afectadas, proporcionando alertas tempranas precisas y confiables que permitan tomar medidas preventivas con anticipación.

En el desarrollo del proyecto se considerarán aspectos como la selección de los sensores adecuados, la implementación de algoritmos de procesamiento y análisis de datos, y la integración de tecnologías de comunicación y difusión de alertas. Además, se llevará a cabo una evaluación de la efectividad del sistema en términos de precisión y tiempo de respuesta. Con el desarrollo de este proyecto, se espera contribuir a la prevención de inundaciones y a la protección de la vida y el patrimonio de las personas.

3. Planteamiento del problema

Un sistema de alerta temprana de inundaciones que incluye la recopilación de datos en el contexto de la Industria 4.0 puede proporcionar una serie de beneficios importantes para las empresas y las comunidades. En primer lugar, la recopilación de datos en tiempo real sobre las condiciones meteorológicas y del agua puede ayudar a predecir con mayor precisión la posibilidad de inundaciones.

La tecnología de la Industria 4.0 permite la integración de múltiples fuentes de datos, como sensores de nivel de agua, estaciones meteorológicas y modelos hidrológicos, lo que permite una comprensión más detallada de las condiciones en tiempo real. Esto puede ayudar a las empresas a tomar decisiones más informadas sobre la gestión de riesgos de inundaciones, como la planificación de evacuaciones, el cierre de operaciones y la preparación de medidas de protección para reducir el impacto en la propiedad y la seguridad de las personas.[1]

En segundo lugar, un sistema de alerta temprana de inundaciones que incluye la recopilación de datos puede ayudar a mejorar la resiliencia de las empresas ante los desastres naturales. Las inundaciones pueden interrumpir las operaciones y provocar daños materiales, pero la preparación adecuada puede minimizar estos efectos. La recopilación de datos en tiempo real y la utilización de tecnologías digitales, como la inteligencia artificial y el aprendizaje automático, pueden ayudar a las empresas a anticipar los riesgos y desarrollar planes de contingencia más efectivos.[2]

En tercer lugar, la recopilación de datos en tiempo real también puede mejorar la eficiencia operativa de las empresas. Al monitorear las condiciones meteorológicas y del agua, las empresas pueden anticipar las condiciones adversas y tomar medidas preventivas para reducir el riesgo de daños y pérdidas. Además, la recopilación de datos también puede ayudar a las empresas a identificar áreas donde se pueden realizar mejoras operativas, como la planificación de la producción y la gestión de la cadena de suministro.[3]

En cuarto lugar, un sistema de alerta temprana de inundaciones que incluye la recopilación de datos puede ayudar a las comunidades a prepararse mejor para los desastres naturales. La recopilación de datos en tiempo real y la utilización de tecnologías digitales pueden ayudar a las comunidades a anticipar los riesgos de inundaciones y desarrollar planes de contingencia más efectivos. Además, la recopilación de datos también puede ayudar a las comunidades a identificar áreas de alto riesgo y tomar medidas preventivas para reducir el impacto en la propiedad y la seguridad de las personas.[4]

4. Justificación del problema

Un sistema de alerta temprana de inundaciones que incluye la recopilación de datos en el contexto de la Industria 4.0 puede proporcionar una serie de beneficios importantes para las empresas y las comunidades. La recopilación de datos en tiempo real sobre las condiciones meteorológicas y del agua permite una comprensión más detallada de las condiciones actuales, lo que ayuda a predecir con mayor precisión la posibilidad de inundaciones.

La tecnología de la Industria 4.0 facilita la integración de múltiples fuentes de datos, como sensores de nivel de agua, estaciones meteorológicas y modelos hidrológicos, lo que proporciona una visión completa de las condiciones en tiempo real. Esto permite a las empresas tomar decisiones más informadas sobre la gestión de riesgos de inundaciones, como la planificación de evacuaciones, el cierre de operaciones y la preparación de medidas de protección para reducir el impacto en la propiedad y la seguridad de las personas.

Además, un sistema de alerta temprana de inundaciones que incluye la recopilación de datos ayuda a mejorar la resiliencia de las empresas ante los desastres naturales. Las inundaciones pueden interrumpir las operaciones y causar daños materiales, pero la preparación adecuada puede minimizar estos efectos. La recopilación de datos en tiempo real y el uso de tecnologías digitales, como la inteligencia artificial y el aprendizaje automático, permiten a las empresas anticipar los riesgos y desarrollar planes de contingencia más efectivos.

La eficiencia operativa también se ve mejorada gracias a la recopilación de datos en tiempo real. Al monitorear las condiciones meteorológicas y del agua, las empresas pueden anticipar las condiciones adversas y tomar medidas preventivas para reducir el riesgo de daños y pérdidas. Además, la recopilación de datos también ayuda a identificar áreas donde se pueden realizar mejoras operativas, como la planificación de la producción y la gestión de la cadena de suministro, lo que conduce a una mayor eficiencia operativa en general.

No solo las empresas se benefician de este sistema, sino que las comunidades también pueden prepararse mejor para los desastres naturales. La recopilación de datos en tiempo real y el uso de tecnologías digitales permiten a las comunidades anticipar los riesgos de inundaciones y desarrollar planes de contingencia más efectivos. Además, la recopilación de datos ayuda a las comunidades a identificar áreas de alto riesgo y tomar medidas preventivas para reducir el impacto en la propiedad y la seguridad de las personas.

Realizando una debida síntesis de lo anterior, un sistema de alerta temprana de inundaciones que incluye la recopilación de datos en el contexto de la Industria 4.0 proporciona beneficios importantes tanto para las empresas como para las comunidades. Estos beneficios incluyen una mayor precisión en la predicción de inundaciones, mejora de la resiliencia empresarial, eficiencia operativa mejorada y una preparación más efectiva para desastres naturales. Gracias a esta tecnología, es posible mitigar los riesgos y minimizar los impactos negativos de las inundaciones tanto en el ámbito empresarial como en el comunitario.

5. Objetivos

5.1. General

Desarrollar y diseñar un sistema de alerta temprana de inundaciones, que permita la identificación oportuna de eventos catastróficos, para mejorar la capacidad de respuesta ante emergencias y reduzca el impacto de los desastres naturales en las comunidades.

5.2. Especifico

- Recopilar datos de sensores de nivel de agua, humedad y temperatura.
- Procesar los datos en tiempo real utilizando una Raspberry Pi para tomar decisiones en tiempo real y detectar posibles inundaciones.
- Transmitir los datos recopilados y las decisiones tomadas a un servidor de datos o a una base de datos en la nube.
- Analizar los datos recopilados mediante algoritmos de análisis de datos para detectar patrones y predecir posibles inundaciones.
- Enviar alertas a los usuarios mediante diferentes medios, como alarmas sonoras o notificaciones en aplicaciones móviles.
- Optimizar el sistema de alerta temprana para aumentar la precisión y la velocidad de detección de posibles inundaciones.
- • Proporcionar una plataforma de monitoreo y análisis de datos en tiempo real para que las autoridades puedan tomar decisiones rápidas y efectivas en situaciones de emergencia.

6. Marco Teórico

Definición de un Sistema de Alerta Temprana de Inundaciones: Un Sistema de Alerta Temprana de Inundaciones (SATI) es un conjunto de herramientas, tecnologías, procedimientos y protocolos diseñados para monitorear y predecir las condiciones hidrológicas y meteorológicas que pueden conducir a inundaciones. El SATI integra múltiples fuentes de datos, como estaciones de monitoreo de nivel de agua, sensores de lluvia, imágenes satelitales, modelos hidrológicos y meteorológicos, y sistemas de comunicación, para proporcionar advertencias anticipadas y oportunas sobre el riesgo de inundaciones.

Componentes de un Sistema de Alerta Temprana de Inundaciones:

Monitoreo y Adquisición de Datos: El SATI requiere una red de estaciones de monitoreo distribuidas estratégicamente para recolectar datos en tiempo real sobre variables hidrológicas y meteorológicas clave, como el nivel de agua en ríos y embalses, la precipitación y la velocidad del viento. Estos datos se transmiten a centros de control para su procesamiento y análisis.

Modelado y Pronóstico: Los modelos hidrológicos y meteorológicos se utilizan para simular el comportamiento de las cuencas hidrográficas y predecir la evolución de los eventos de inundación. Estos modelos integran datos históricos, datos en tiempo real y pronósticos meteorológicos para estimar los volúmenes de agua, las áreas de inundación y los tiempos de llegada del pico de inundación.

Sistemas de Alerta y Comunicación: Los sistemas de alerta se basan en los resultados del monitoreo y pronóstico para emitir advertencias y avisos de inundación. Estas alertas se envían a través de diferentes canales de comunicación, como mensajes de texto, sirenas, aplicaciones móviles y medios de comunicación masiva, para alcanzar a la población en riesgo y a las autoridades responsables de la gestión de desastres.

Capacidades de Respuesta y Planificación: Un SATI efectivo incluye la definición de protocolos de respuesta y planes de evacuación en caso de inundaciones. Estos planes deben estar respaldados por

una coordinación efectiva entre las autoridades locales, regionales y nacionales, así como por la participación activa de la comunidad en la implementación de las medidas de preparación y respuesta.

Beneficios y Desafíos de los Sistemas de Alerta Temprana de Inundaciones:

Beneficios:

- Reducción del riesgo de pérdidas humanas y daños materiales.
- Mayor tiempo para tomar medidas de precaución y evacuación.
- Mejora de la capacidad de respuesta de los servicios de emergencia.
- Mayor conciencia y conocimiento de la población sobre los riesgos de inundación.
- Optimización de la planificación y gestión de la infraestructura hidráulica.

Desafíos:

- Disponibilidad y calidad de los datos hidrológicos y meteorológicos.
- Implementación y mantenimiento de una red de monitoreo adecuada.
- Precisión y confiabilidad de los modelos y pronósticos.
- Desarrollo de sistemas de alerta y comunicación efectivos.
- Coordinación entre diferentes entidades y niveles de gobierno.

Ejemplos de Sistemas de Alerta Temprana de Inundaciones:

1. **Sistema de Alerta Temprana de Inundaciones del río Mekong:** El río Mekong es uno de los principales ríos de Asia, que atraviesa varios países, incluidos China, Myanmar, Laos, Tailandia, Camboya y Vietnam. Debido a su extenso alcance y a las fuertes precipitaciones en la región, el río Mekong es propenso a las inundaciones, que pueden causar graves daños a la vida y la propiedad.

El Sistema de Alerta Temprana de Inundaciones del río Mekong es un proyecto regional que se ha implementado en colaboración entre los países mencionados. El objetivo principal del sistema es proporcionar advertencias oportunas sobre las inundaciones a lo largo del río Mekong y sus afluentes, permitiendo la toma de medidas preventivas y de respuesta adecuadas.

El sistema utiliza una combinación de tecnologías, incluidos los datos satelitales, modelos hidrológicos y meteorológicos, y una red de estaciones de monitoreo distribuidas a lo largo del río y sus afluentes. Estas estaciones miden el nivel del agua, la precipitación y otros parámetros relevantes para evaluar el riesgo de inundación.

La información recopilada se procesa y analiza en centros de control, donde se generan pronósticos y alertas de inundación. Estas alertas se envían a través de diferentes canales de comunicación, como mensajes de texto, sirenas y aplicaciones móviles, para llegar a las comunidades y autoridades locales en riesgo. También se establecen protocolos de respuesta y se lleva a cabo la coordinación entre los países para garantizar una respuesta efectiva en caso de inundación.

2. **Sistema de Alerta Temprana de Inundaciones de los Países Bajos:** Los Países Bajos, conocidos por su gran parte de tierras bajas y la presencia de sistemas de diques y compuertas, han desarrollado uno de los sistemas de alerta temprana de inundaciones más avanzados del mundo. Dado que gran parte del territorio del país está por debajo del nivel del mar, la prevención y gestión de las inundaciones son fundamentales.

El Sistema de Alerta Temprana de Inundaciones de los Países Bajos combina tecnología, infraestructura y planificación eficientes para prevenir y responder a las inundaciones. Algunos aspectos destacados del sistema incluyen:

Red de Monitoreo: Los Países Bajos cuentan con una extensa red de estaciones de monitoreo distribuidas en ríos, diques y embalses. Estas estaciones miden constantemente los niveles de agua y otras variables hidrológicas clave.

Modelado y Pronóstico: Se utilizan modelos hidrológicos y meteorológicos para simular el comportamiento de los ríos y estimar el aumento del nivel del agua en caso de eventos extremos. Los pronósticos se actualizan regularmente para mantenerse al tanto de las condiciones cambiantes.

Infraestructura de Defensa: Los Países Bajos han construido una extensa red de diques, compuertas y sistemas de drenaje para proteger las áreas vulnerables. Estos sistemas se monitorean y mantienen regularmente para garantizar su efectividad.

Comunicación y Alerta: El sistema utiliza diferentes canales de comunicación para enviar alertas a la población en riesgo. Se utilizan mensajes de texto, sirenas, medios de comunicación y aplicaciones móviles para informar a las personas y brindar instrucciones sobre cómo actuar.

Planificación de Emergencias: Los Países Bajos tienen planes de emergencia detallados y protocolos de respuesta para abordar las situaciones de inundación. Se realizan simulacros y ejercicios regulares para garantizar la preparación y la coordinación adecuada entre los diferentes actores.

Puntos clave a tener en cuenta:

1. **Cambio climático y eventos extremos:** El cambio climático es un factor importante que contribuye al aumento de eventos climáticos extremos, incluyendo inundaciones. A medida que aumenta la concentración de gases de efecto invernadero en la atmósfera, se producen cambios en los patrones climáticos y se intensifican los fenómenos meteorológicos. Algunos de los aspectos relevantes a considerar son:
 - **Patrones de precipitación:** El cambio climático puede alterar la distribución de la precipitación, lo que significa que algunas regiones pueden experimentar períodos más secos, mientras que otras pueden enfrentar lluvias más intensas y frecuentes. Estos cambios en los patrones de precipitación pueden aumentar el riesgo de inundaciones en áreas propensas a fuertes lluvias.
 - **Nivel del mar:** El aumento de las temperaturas globales provoca el derretimiento de los glaciares y la expansión térmica de los océanos, lo que resulta en un aumento del nivel del mar. Esto hace que las áreas costeras sean más vulnerables a las inundaciones costeras y aumenta la intrusión de agua salada en los sistemas de agua dulce.
 - **Fenómenos meteorológicos extremos:** El cambio climático también puede influir en la frecuencia e intensidad de fenómenos meteorológicos extremos, como tormentas tropicales y huracanes. Estos eventos pueden desencadenar fuertes lluvias y aumentar el riesgo de inundaciones repentinas.
2. **Participación comunitaria:** La participación comunitaria es fundamental para el éxito de los sistemas de alerta temprana de inundaciones. Algunos aspectos relevantes a considerar son:
 - **Educación sobre riesgos:** La educación y la concientización son esenciales para que las comunidades comprendan los riesgos asociados con las inundaciones y cómo responder de

manera adecuada. Esto incluye la difusión de información sobre las causas de las inundaciones, las señales de alerta y los planes de evacuación.

- **Desarrollo de capacidades locales:** Capacitar a las comunidades locales en la interpretación de la información de alerta temprana, el establecimiento de sistemas de comunicación eficientes y la implementación de medidas de preparación puede mejorar su capacidad de respuesta y reducir el impacto de las inundaciones.
- **Participación en la toma de decisiones:** Involucrar a las comunidades en la planificación y toma de decisiones relacionadas con la gestión del riesgo de inundaciones puede ayudar a asegurar que se tengan en cuenta las necesidades y preocupaciones locales. Esto puede incluir la creación de comités de gestión de desastres y la participación en simulacros y ejercicios de preparación.

3. **Gestión del riesgo de inundaciones:** La gestión del riesgo de inundaciones implica una serie de medidas destinadas a reducir la vulnerabilidad de las comunidades y los activos frente a las inundaciones. Algunos aspectos relevantes a considerar son:

- **Identificación de peligros:** Evaluar los peligros de inundación en una región determinada, incluyendo la probabilidad de que ocurran inundaciones y los posibles escenarios de inundación.
- **Evaluación de vulnerabilidades:** Determinar la vulnerabilidad de las comunidades, infraestructuras y ecosistemas frente a las inundaciones. Esto implica evaluar factores como la ubicación, la exposición y la capacidad de resistencia de los elementos vulnerables.
- **Planificación de la infraestructura de defensa:** Desarrollar e implementar medidas de ingeniería, como diques, barreras y sistemas de drenaje, para reducir el riesgo de inundaciones y proteger las áreas más vulnerables.
- **Estrategias de mitigación:** Adoptar medidas de mitigación, como la conservación y restauración de los ecosistemas naturales, la ordenación del territorio para evitar la construcción en áreas de alto riesgo y la implementación de políticas de gestión del agua.

4. **Impacto socioeconómico de las inundaciones:** Las inundaciones tienen impactos significativos en diversos aspectos. Algunos puntos a considerar son:

- **Impacto en la salud y la seguridad:** Las inundaciones pueden causar lesiones y pérdidas de vidas humanas. Además, pueden generar problemas de salud, como enfermedades transmitidas por el agua y brotes de enfermedades infecciosas.
- **Impacto en la infraestructura:** Las inundaciones pueden dañar y destruir infraestructuras clave, como carreteras, puentes, edificios y sistemas de suministro de agua y energía. Esto puede tener un impacto duradero en el desarrollo económico de una región.
- **Impacto en la economía:** Las inundaciones pueden interrumpir las actividades económicas, como la agricultura, la industria y el comercio. Además, las pérdidas de infraestructura y propiedades pueden tener un impacto negativo en la economía local y nacional.
- **Impacto ambiental:** Las inundaciones pueden causar daños significativos a los ecosistemas, como la degradación de los hábitats acuáticos, la pérdida de biodiversidad y la contaminación del agua.

7. Marco Experimental

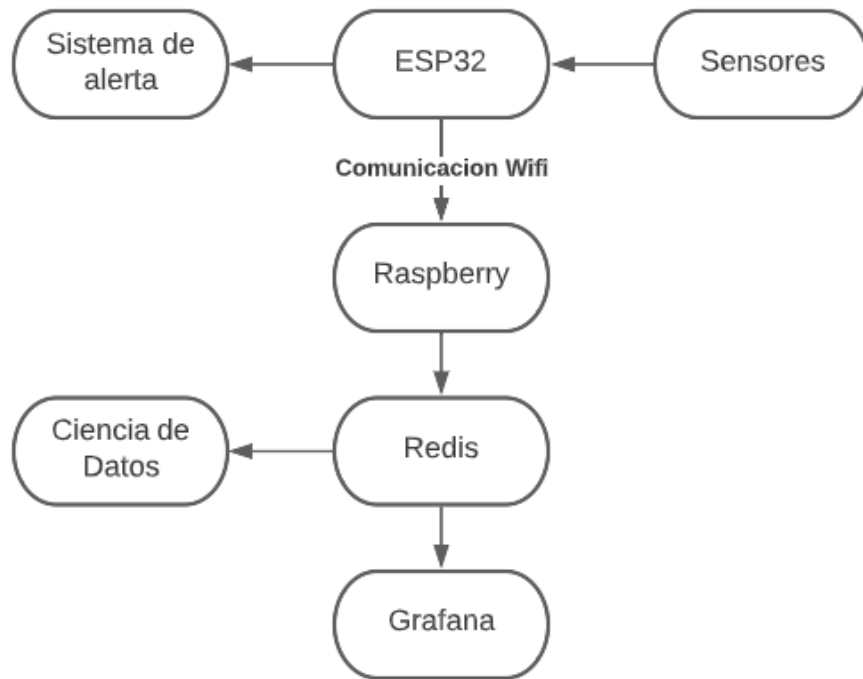


Figura 1. Diagrama de Bloques.

7.1. Recopilación de datos, análisis de sensores y actuadores

Mediante la ESP32 como un sistema IOT, se hace uso de su conexión WiFi para recopilar datos y posteriormente enviarlos con un broker MQTT. Para realizar esta tarea la ESP32 se programa en Arduino.

Inicialmente, se programan los sensores utilizados, para el desarrollo de este proyecto se hizo uso de el sensor de ultrasonido HC-SR04 y el sensor SHT30 de temperatura y humedad. El sensor de ultrasonido cuenta con 4 pines, donde VCC y GND son la alimentación, mientras el pin TRIG se encarga de enviar un pulso en el transmisor y el pin ECHO se encarga de recibir la onda de sonido al devolverse. De esta forma, se logra obtener el dato de la distancia de acuerdo al tiempo que se demoró en recibir la señal el pin ECHO.



Figura 2. Sensor de ultrasonido HC-SR04.

En la programación en Arduino se crea una función que se encarga de leer la distancia, y retorna esta variable. Para la lectura del sensor se usa la librería *Ultrasonic.h* y posteriormente se usa la sentencia *ultrasonic.read()* para la lectura de la información y se guarda en la variable distancia. Debido a que este sensor medirá el nivel del agua, se maneja un condicional donde al alcanzar una distancia se alerta

mediante un buzzer del crecimiento en el agua. Por último, se retorna esta variable como flotante.

```
float ultrasonido() {  
    float distancia = ultrasonic.read();  
    distancia = 25 - distancia;  
    if (distancia > 15){digitalWrite(BUZZER, HIGH);}  
    else {digitalWrite(BUZZER, LOW);}  
    Serial.print("Distancia: ");  
    Serial.print(distancia);  
    Serial.println(" cm");  
    return distancia;  
}
```

Figura 3. Función del sensor de ultrasonido.

Ahora para el sensor SHT30, se maneja mediante comunicación i2c por lo que en la ESP32 se conecta a los pines SDA y SCL. No es necesario el uso de resistencias de pull-up debido a que este sensor ya las tiene incluidas internamente. En cuanto a la programación se usaron las librerías *Wire.h* y *Adafruit_SHT31.h* para la configuración de los pines SDA y SCL, la segunda librería se usa para la lectura de la información. Se realizaron dos funciones, la primera se encarga de leer la temperatura y para esto usa la sentencia *sht31.readTemperature()*. La segunda lee la temperatura con la sentencia *sht31.readHumidity()*. Por último, en ambas funciones se retorna como flotante el valor de la temperatura y la humedad respectivamente.

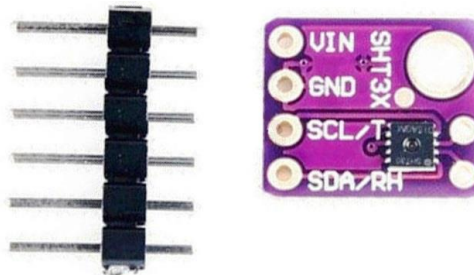


Figura 4. Sensor SHT30.

```
float temp() {  
    float temperatura = sht31.readTemperature();  
    Serial.print("Temperatura: ");  
    Serial.print(temperatura);  
    Serial.println(" °C");  
    return temperatura;  
}
```

Figura 5. Función temperatura.

```
float hum() {  
    float humedad = sht31.readHumidity();  
    Serial.print(" Humedad: ");  
    Serial.print(humedad);  
    Serial.println(" %");  
    return humedad;  
}
```

Figura 6. Función humedad.

Una vez hechas las funciones que leen las variables recopiladas mediante los sensores, procedemos a realizar las funciones que se encargan de la conexión WiFi. Para esto lo primero es incluir dos librerías más, WiFi.h y PubSubClient.h. Ahora se inicializan unas constantes para la conexión de la red y al broker MQTT.

```
//const char* ssid = "Omni_lite611D20";  
//const char* password = "major8954";  
const char* ssid = "Romero";  
const char* password = "maniva01";  
//const char* ssid = "iPhone de Nicolas";  
//const char* password = "mendez10";  
const char* mqtt_server = "broker.mqttdashboard.com";  
  
WiFiClient espClient;  
PubSubClient client(espClient);
```

Figura 7. Constantes para la conexión WiFi.

Ahora, se crea la función *setup_wifi*, se encarga de la configuración y conexión a la red WiFi. Luego, se realiza la función *callback* usada para que se pueda enviar los datos recopilados. Por último, está la función *reconnect*, que como su nombre indica se encargara de la reconexión a la red WiFi dado el caso se haya perdido esta conexión.

```
void setup_wifi() {  
    WiFi.begin(ssid,password);  
    // Mientras el ESP32 no se conecte al AP:  
    while(WiFi.status() != WL_CONNECTED) {  
        delay(200);  
        Serial.print(".");  
    }  
}
```

Figura 8. Función *setup_wifi*.

```
void callback(char* topic, byte* payload, unsigned int length) {  
    payload[length]='\0';  
    String val = String((char*)payload);  
}
```

Figura 9. Función *callback*.

```
void reconnect() {  
    if(client.connect("ESP32")) {  
        Serial.println(".....Conexión exitosa");  
    }  
}
```

Figura 10. Función *reconnect*.

Una vez creadas todas las funciones necesarias, ahora si se programa el *void setup* y *void loop* de Arduino. Primero en el *setup*, se inicializa el serial y wire, se declara el pin del buzzer para la alerta como una salida, continuamos indicando la dirección del sensor para la comunicación. Y, por último, se llama función *setup_wifi* y se programa cual será el broker y callback.

```

void setup() {
  Serial.begin(115200);
  Wire.begin();
  pinMode(BUZZER, OUTPUT);

  if (!sht31.begin(0x44)) { // Inicializar el sensor SHT31
    Serial.println("No se pudo encontrar el sensor SHT31");
    while (1);
  }
  setup_wifi();
  client.setServer(mqtt_server,1883);
  client.setCallback(callback);
}

```

Figura 11. void setup.

Para finalizar, se programa el loop que se encarga de repetir el código siempre. Acá, primero revisamos la conexión, dado el caso se haya perdido se llama la función *reconnect* ahora guardamos las variables de temperatura, humedad y distancia usando las funciones anteriormente creadas. Ya a lo último enviamos los datos indicando que serán en el topic *canal*.

```

void loop() {
  if(!client.connected()){reconnect();}
  client.loop();
  temperatura = temp();
  humedad = hum();
  distancia = ultrasonido();
  sprintf(msg1,20,"%0.2f,%0.2f,%0.2f",temperatura,humedad,distancia);
  client.publish("canal",msg1);
  delay(100);
}

```

Figura 12. void loop.

7.2. Envío de datos a treves de comunicación wifi

Para la recepción de los datos se hará uso de Node Red. En esto es necesario un broker, en esta ocasión se trabajó con HiveMQ Cloud, en donde se creó el servidor y para Node Red se configuro usando los nodos Mqtt In y Mqtt Out.

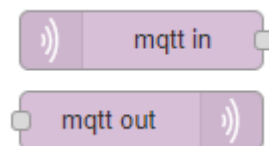
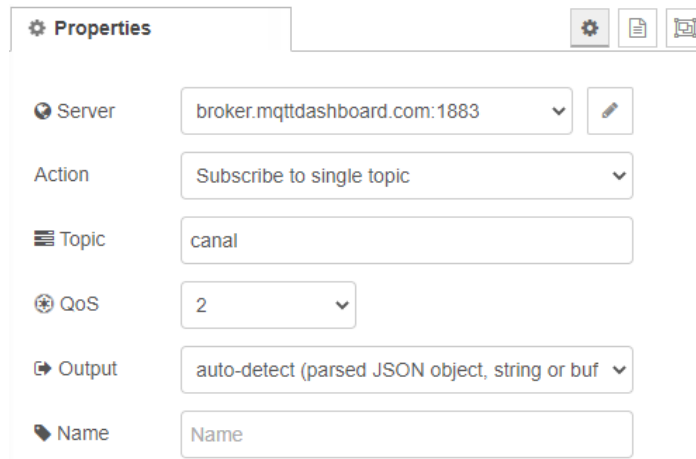


Figura 13. Nodos Mqtt.

Para los nodos Mqtt es necesario especificar el broker al que se uno se requiere conectar, en este caso también fue necesario indicar usuario y contraseña para habilitar la conexión.



Properties

Server: broker.mqttdashboard.com:1883

Action: Subscribe to single topic

Topic: canal

QoS: 2

Output: auto-detect (parsed JSON object, string or buf)

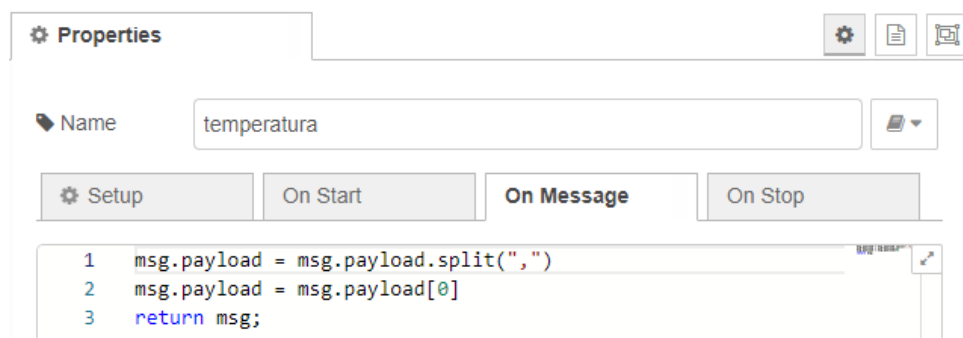
Name: Name

Figura 14. Configuración nodos Mqtt.

Ahora, ya hemos recibido los datos en Node Red, solamente que se recibieron los tres datos en una sola cadena. La cadena separa los datos con comas por lo tanto se usó el nodo función para separar todos estos datos en distintas variables y posteriormente visualizarlas.



Figura 15. Nodo Función.



Properties

Name: temperatura

Setup On Start **On Message** On Stop

```

1 msg.payload = msg.payload.split(",")
2 msg.payload = msg.payload[0]
3 return msg;

```

Figura 16. Código para separar los datos.

Una vez han sido separados los datos, se procede a usar los nodos gauge y led para el dashboard. También se usa un nodo Mqtt Out con su respectivo topic, de acuerdo a las tres variables.

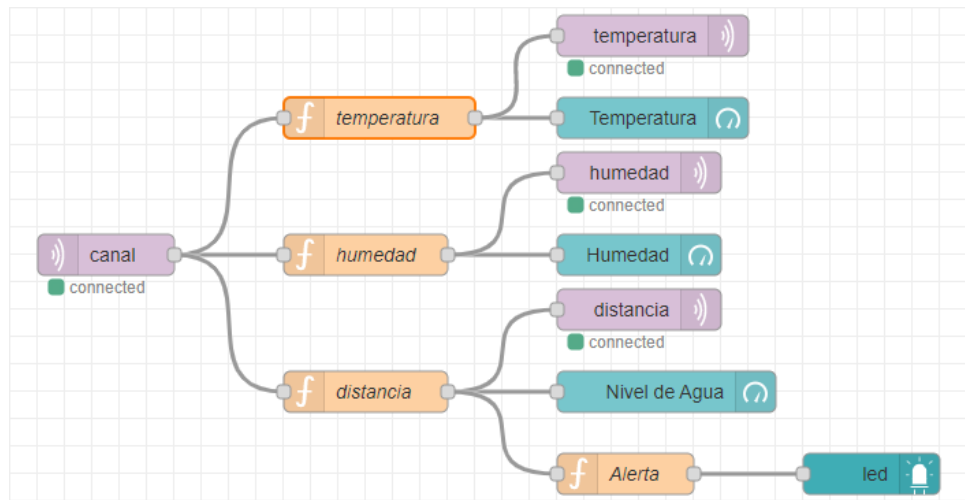


Figura 17. Flow Node Red.

Ya declarados los Mqtt Out se continua en la raspberry para recibir los datos.

7.3. Recepción de datos a través de la Raspberry

En la raspberry se hace uso de Geany como editor de texto para Python. Lo primero es importar las librerías requeridas. Para recibir los datos que envía Node Red, se usa la librería *paho.mqtt.client*, además importamos *redis* y *time*.

```
import paho.mqtt.client as mqtt
import redis
import time
```

Figura 18. Librerías importadas.

Ahora, se indica la base de datos en redis a la que nos deseamos comunicar. Para esto se usó Redis Cloud Console o también se conoce como Redis Labs.

Status	Name	Endpoint	Memory	Plan Usage	Modules	Options
●	Nicolas-free-db	Connect	2.4MB	8.2%	Redis Stack	

Items per page 10 << 1 of 1 >> Page 1 of 1

Figura 19. Redis Cloud Console.

```
r = redis.Redis(
    host = 'redis-10258.c60.us-west-1-2.ec2.cloud.redislabs.com',
    port=10258,
    password='BYKu0Y2n4cNsBqIrGE08kDwUhBbhYNR6')
```

Figura 20. Puerto Redis.

Se continuó inicializando las variables en donde se almacenarán los datos.

```
temperatura = ""
humedad = ""
distancia = ""
```

Figura 21. Inicialización de variables.

Posteriormente se crea la función *on_message*. Se declaran como globales las variables temperatura, humedad y distancia, y se crean las variables mensaje y tema donde se guardarán la información de los sensores y el topic respectivamente. Ahora como se recibe el mensaje como un string se convierte el dato en un flotante y se guarda en la variable valor.

```
def on_message(client, userdata, msg):
    global temperatura, humedad, distancia

    mensaje = msg.payload.decode()
    tema = msg.topic

    try:
        valor = float(mensaje)
    except ValueError:
        print(f"Error: el valor recibido en el tema '{tema}' no es numérico")
    return
```

Figura 22. Recopilación información en la función on_message.

Continuando con la función *on_message* se crea un objeto Time Series en redis, y se crean en Redis cada una de las keys requeridas.

```
ts = r.ts()
#ts.create("temperatura")
#ts.create("humedad")
#ts.create("distancia")
```

Figura 23. Creación de las keys en Redis.

Luego, de acuerdo a cada uno de los topic que se envían desde Node Red dependiendo cual sea se guarda el valor en su respectiva variable. A su vez, estos datos se añaden a sus respectivas keys en redis. Por último en la función, se imprimen los datos recibidos.

```
if tema == "temperatura":
    temperatura = valor
    ts.add("temperatura", "*", temperatura)
elif tema == "humedad":
    humedad = valor
    ts.add("humedad", "*", humedad)
elif tema == "distancia":
    distancia = valor
    ts.add("distancia", "*", distancia)

print("Temperatura:", temperatura)
print("Humedad:", humedad)
print("Distancia:", distancia)
print("")
```

Figura 24. Envío de datos a Redis.

Al salir de la función, es necesario conectarse al broker y suscribirse a cada uno de los topic enviados desde Node Red. Al final se usa la sentencia *loop_start* para que se repita y se mantenga la conexión.

```
broker_address = "broker.mqttdashboard.com"
client = mqtt.Client()
client.on_message = on_message

client.connect(broker_address)
client.subscribe("temperatura")
client.subscribe("humedad")
client.subscribe("distancia")

client.loop_start()
```

Figura 25. Conexión al broker.

7.4. Ciencia de datos a partir de Python

Python es un lenguaje de programación ampliamente utilizado en el campo de la ciencia de datos debido a su flexibilidad y a las numerosas bibliotecas disponibles. Para recibir datos de Redis en Python, se utilizaron bibliotecas como redis-py para establecer una conexión con la base de datos Redis y realizar operaciones de lectura.

```
temp = ts.get("temperatura", True)
print(temp)
hum = ts.get("humedad", True)
print(hum)
dist = ts.get("distancia", True)
print(dist)
```

Figura 26. Código para la obtención de las variables

Una vez establecida la conexión, se pueden recuperar los datos almacenados en Redis utilizando los comandos correspondientes, como lrange para obtener una lista de valores. Estos datos pueden representar variables como temperatura, humedad y nivel de agua. Una vez que los datos se han obtenido de Redis, es posible crear un archivo CSV utilizando Python para almacenarlos de manera estructurada. Esto se puede hacer utilizando el módulo csv, que permite escribir los datos en un archivo CSV con formato adecuado.

```
# Crear DataFrame vacío
df = pd.DataFrame(columns=['Fecha', 'Hora', 'Temperatura', 'Nivel_agua', 'Humedad'])
```

Figura 27. Creación del dataframe(csv)

El DataFrame se inicializa con columnas especificadas por los nombres 'Fecha', 'Hora', 'Temperatura', 'Nivel_agua' y 'Humedad'. Estas columnas representan las variables o atributos que se desean almacenar en el DataFrame. El parámetro "columns" se utiliza para especificar los nombres de las columnas y crea un DataFrame con estas columnas vacías. El DataFrame resultante será una estructura de datos tabular con columnas tituladas como se especificó.

Una vez que los datos se han almacenado en el archivo CSV, se pueden utilizar diversas bibliotecas de visualización de datos, como matplotlib, para crear gráficas que representen las variables de interés. En este caso, se generaron gráficas individuales para cada variable, como temperatura, humedad y nivel de agua, en función del tiempo.


```
# Configuración de las subplots
ax1 = fig.add_subplot(3, 2, 1)
ax2 = fig.add_subplot(3, 2, 3)
ax3 = fig.add_subplot(3, 2, 5)
ax4 = fig.add_subplot(3, 2, 2)
ax5 = fig.add_subplot(3, 2, 4)
ax6 = fig.add_subplot(3, 2, 6)
```

Figura 28. Configuración de los subplot

```
def update_plots(i):
    ax1.clear()
    ax2.clear()
    ax3.clear()
    ax4.clear()
    ax5.clear()
    ax6.clear()

    ax1.plot(df.index, df['Nivel_agua'], label='Nivel_agua')
    ax1.set_title('Nivel de agua')
    ax1.set_xlabel('Fecha y Hora')
    ax1.set_ylabel('Nivel')
    ax1.legend()

    ax2.plot(df.index, df['Humedad'], label='Humedad')
    ax2.set_title('Humedad')
    ax2.set_xlabel('Fecha y Hora')
    ax2.set_ylabel('Humedad')
    ax2.legend()

    ax3.plot(df.index, df['Temperatura'], label='Temperatura')
    ax3.set_title('Temperatura')
    ax3.set_xlabel('Fecha y Hora')
    ax3.set_ylabel('Temperatura')
    ax3.legend()
```

Figura 29. Función para actualizar las gráficas en tiempo real

Además de la visualización, Python ofrece una amplia gama de bibliotecas para el análisis de datos, como Pandas y NumPy. Estas bibliotecas permiten realizar operaciones estadísticas y de manipulación de datos en los conjuntos de datos obtenidos. Como por ejemplo “seasonal_decompose” que realiza una descomposición de la serie temporal en sus componentes: tendencia, estacionalidad y residuos.

```
if len(df) >= 14:
    nivel_agua_decompose = seasonal_decompose(df['Nivel_agua'], model='additive', per:
    humedad_decompose = seasonal_decompose(df['Humedad'], model='additive', period=7)
    temperatura_decompose = seasonal_decompose(df['Temperatura'], model='additive', pe
```

Figura 30. Definición del if

Este “if” se emplea ya que la función `seasonal_descompose` necesita por lo menos 14 datos que busca en el “csv” y que se van actualizando en tiempo real para poder hacer el correcto análisis

```
ax4.plot(df.index, nivel_agua_decompose.trend, label='Tendencia')
ax4.plot(df.index, nivel_agua_decompose.seasonal, label='Estacionalidad')
ax4.plot(df.index, nivel_agua_decompose.resid, label='Residuos')
ax4.set_title('Descomposición - Nivel de agua')
ax4.set_xlabel('Fecha y Hora')
ax4.legend()
```

Figura 31. aplicación de la descomposición para el nivel de agua

La primera línea de código `ax4.plot(df.index, nivel_agua_decompose.trend, label='Tendencia')` traza la tendencia de la serie temporal de nivel de agua en función del índice de tiempo (por ejemplo, fechas o marcas de tiempo) en el eje x y los valores de la tendencia en el eje y. La etiqueta 'Tendencia' se utiliza para identificar esta línea en la leyenda del gráfico.

La segunda línea de código `ax4.plot(df.index, nivel_agua_decompose.seasonal, label='Estacionalidad')` traza la estacionalidad de la serie temporal, que representa los patrones periódicos o repetitivos a lo largo del tiempo. Al igual que en la línea anterior, se utilizan los índices de tiempo en el eje x y los valores de la estacionalidad en el eje y. La etiqueta 'Estacionalidad' se utiliza para identificar esta línea en la leyenda del gráfico.

La tercera línea de código `ax4.plot(df.index, nivel_agua_decompose.resid, label='Residuos')` traza los residuos de la serie temporal, que son las componentes no explicadas por la tendencia y la estacionalidad. Los índices de tiempo se utilizan en el eje x y los valores de los residuos en el eje y. La etiqueta 'Residuos' se utiliza para identificar esta línea en la leyenda del gráfico.

```
# Guardar en CSV
df.to_csv('datos.csv', index=False)

yield

# Generar los datos aleatorios y actualizar las gráficas en tiempo real
ani = animation.FuncAnimation(fig, update_plots, frames=generate_data, interval=1000)

plt.show()
```

Figura 32. Guardado del CSV y actualización de las graficas

Por último, guardamos los datos en el csv para posteriormente plotear todos los datos que corresponden.

Luego de realizar la creación del csv entra en juego la creación de una nueva interfaz gráfica la cual nos permite la búsqueda de datos a partir de varias condiciones.

```
def buscar_maximo_nivel_agua():|
    # Leer el archivo CSV
    df = pd.read_csv('datos.csv')

    # Obtener el índice del valor máximo de nivel de agua
    idx_max_nivel_agua = df['Nivel_agua'].idxmax()

    # Obtener la fila correspondiente al índice máximo
    fila_max_nivel_agua = df.loc[idx_max_nivel_agua]

    # Mostrar el resultado en una ventana emergente
    messagebox.showinfo('Datos a partir del CSV', f'El valor máximo de Nivel de Agua es:\n\n{fila_max_nivel_agua}')
```

Figura 33. Definición para buscar el mayor dato del Nivel del agua

```
def buscar_maxima_temperatura():
    # Leer el archivo CSV
    df = pd.read_csv('datos.csv')

    # Obtener el índice del valor máximo de temperatura
    idx_max_temperatura = df['Temperatura'].idxmax()

    # Obtener la fila correspondiente al índice máximo
    fila_max_temperatura = df.loc[idx_max_temperatura]

    # Mostrar el resultado en una ventana emergente
    messagebox.showinfo('Datos a partir del CSV', f'El valor máximo de Temperatura es:\n\n{fila_max_temperatura}')
```

Figura 34. Definición para buscar el mayor dato de Temperatura

```
def buscar_maxima_humedad():
    # Leer el archivo CSV
    df = pd.read_csv('datos.csv')

    # Obtener el índice del valor máximo de humedad
    idx_max_humedad = df['Humedad'].idxmax()

    # Obtener la fila correspondiente al índice máximo
    fila_max_humedad = df.loc[idx_max_humedad]

    # Mostrar el resultado en una ventana emergente
    messagebox.showinfo('Datos a partir del CSV', f'El valor máximo de Humedad:\n\n{fila_max_humedad}')
```

Figura 35. Definición para buscar el mayor dato del Humedad

Las funciones `buscar_maximo_nivel_agua`, `buscar_maxima_temperatura` y `buscar_maxima_humedad` se encargan de leer el archivo CSV, encontrar el valor máximo de nivel de agua, temperatura y humedad, respectivamente, y mostrar el resultado en una ventana emergente.

```
def buscar_fecha_hora():
    # Obtener los valores ingresados por el usuario
    fecha = entry_fecha.get()
    hora = entry_hora.get()

    # Leer el archivo CSV
    df = pd.read_csv('datos.csv')

    # Filtrar por fecha y hora
    if fecha and hora:
        filtro_fecha_hora = (df['Fecha'] == fecha) & (df['Hora'] == hora)
        df = df[filtro_fecha_hora]

    # Mostrar los resultados en una ventana emergente
    mostrar_resultados(df)
```

Figura 36. Definición para buscar por Fecha y hora

La función `buscar_fecha_hora` obtiene los valores ingresados por el usuario para la fecha y la hora, lee el archivo CSV y filtra los datos según la fecha y la hora ingresadas. Luego muestra los resultados en una ventana emergente.

```
def buscar_nivel_agua():
    # Obtener los valores ingresados por el usuario
    nivel_agua_min = float(entry_nivel_agua_min.get())
    nivel_agua_max = float(entry_nivel_agua_max.get())

    # Leer el archivo CSV
    df = pd.read_csv('datos.csv')

    # Filtrar por rango de nivel de agua
    filtro_nivel_agua = (df['Nivel_agua'] >= nivel_agua_min) & (df['Nivel_agua'] <= nivel_agua_max)
    df = df[filtro_nivel_agua]
```

Figura 37. Definición para buscar por rangos de nivel de agua

La función `buscar_nivel_agua` obtiene los valores ingresados por el usuario para el nivel de agua mínimo y máximo, lee el archivo CSV y filtra los datos dentro de ese rango. Luego muestra los resultados en una ventana emergente.

```
def buscar_temperatura_humedad_nivel_agua():
    # Obtener los valores ingresados por el usuario
    temperatura_min = float(entry_temperatura_min.get())
    temperatura_max = float(entry_temperatura_max.get())
    humedad_min = float(entry_humedad_min.get())
    humedad_max = float(entry_humedad_max.get())
    nivel_agua_min = float(entry_nivel_agua_min2.get())
    nivel_agua_max = float(entry_nivel_agua_max2.get())

    # Leer el archivo CSV
    df = pd.read_csv('datos.csv')

    # Filtrar por rango de temperatura, humedad y nivel de agua
    filtro_temperatura = (df['Temperatura'] >= temperatura_min) & (df['Temperatura'] <= temperatura_max)
    filtro_humedad = (df['Humedad'] >= humedad_min) & (df['Humedad'] <= humedad_max)
    filtro_nivel_agua = (df['Nivel_agua'] >= nivel_agua_min) & (df['Nivel_agua'] <= nivel_agua_max)

    df = df[filtro_temperatura & filtro_humedad & filtro_nivel_agua]

    # Mostrar los resultados en una ventana emergente
    mostrar_resultados(df)
```

Figura 38. Definición para buscar por rangos de humedad-temperatura y nivel de agua

La función `buscar_temperatura_humedad_nivel_agua` obtiene los valores ingresados por el usuario para la temperatura mínima y máxima, la humedad mínima y máxima, y el nivel de agua mínimo y máximo. Luego lee el archivo CSV y filtra los datos según estos rangos. Finalmente, muestra los resultados en una ventana emergente.

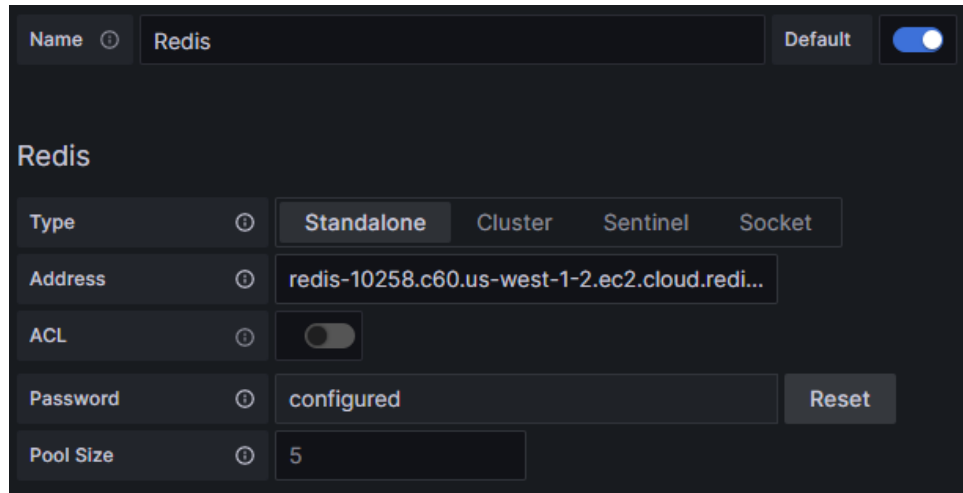
```
def mostrar_resultados(df):
    resultados = tk.Toplevel(root)
    resultados.title('Resultados')

    if df.empty:
        tk.Label(resultados, text='No se encontraron resultados').pack()
    else:
        tk.Label(resultados, text='Resultados:').pack()
        tk.Label(resultados, text=df.to_string()).pack()
```

Figura 39. Función para mostrar los resultados.

7.5. Visualización de la información recopilada

Para la visualización de los datos recopilados se realiza un dashboard en Grafana, para esto lo primero fue declarar la data source en redis de la cual se estarán recibiendo los datos. Para esto se utilizan los mismos datos del Redis Cloud Console.



The image shows the Redis data source configuration interface in Grafana. At the top, there is a 'Name' field set to 'Redis' and a 'Default' toggle switch that is turned on. Below this, the 'Redis' section contains several fields: 'Type' is set to 'Standalone' (with options for Standalone, Cluster, Sentinel, and Socket); 'Address' is 'redis-10258.c60.us-west-1-2.ec2.cloud.redi...'; 'ACL' is a toggle switch that is turned off; 'Password' is 'configured' with a 'Reset' button; and 'Pool Size' is '5'.

Figura 40. Configuración del data source.

Ahora, se crea el dashboard y se agregan tres Row, Datos de la Temperatura y Humedad, Nivel del Agua y Tablas.

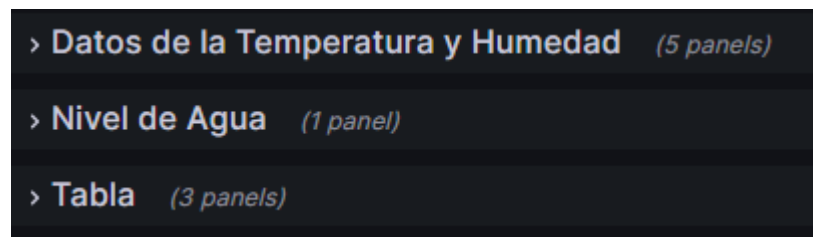
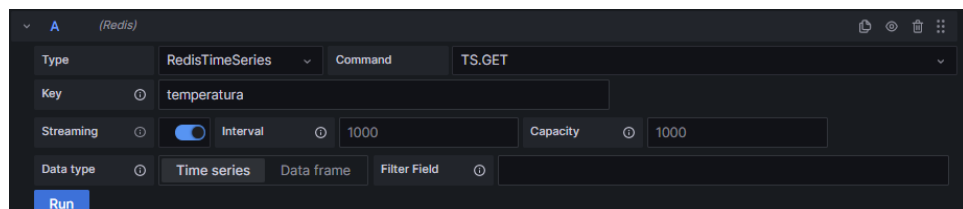


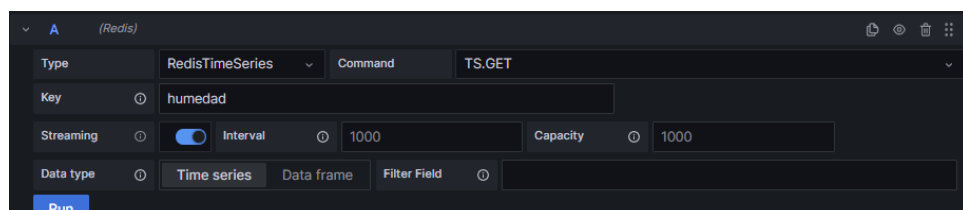
Figura 41. Row creados.

En el primer Row, se crean dos gauge para visualizar los datos de temperatura y humedad.



The image shows the configuration for a RedisTimeSeries data source in Grafana. The 'Type' is 'RedisTimeSeries' and the 'Command' is 'TS.GET'. The 'Key' is 'temperatura'. The 'Streaming' toggle is turned on, with an 'Interval' of 1000 and a 'Capacity' of 1000. The 'Data type' is 'Time series'. There is a 'Run' button at the bottom.

Figura 42. Gauge Temperatura.



The image shows the configuration for a RedisTimeSeries data source in Grafana. The 'Type' is 'RedisTimeSeries' and the 'Command' is 'TS.GET'. The 'Key' is 'humedad'. The 'Streaming' toggle is turned on, with an 'Interval' of 1000 and a 'Capacity' of 1000. The 'Data type' is 'Time series'. There is a 'Run' button at the bottom.

Figura 43. Gauge Humedad.

Ahora, de la misma forma se grafica los datos de Temperatura y Humedad usando el panel Time Series.

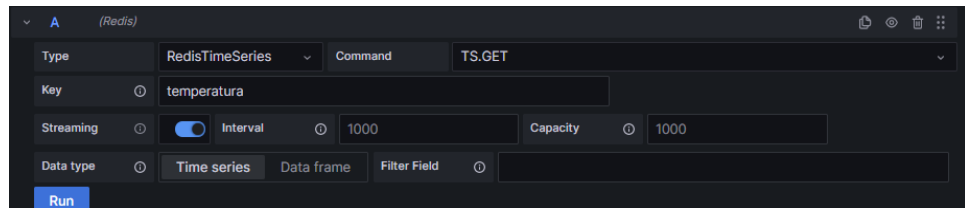


Figura 44. Time Series Temperatura.

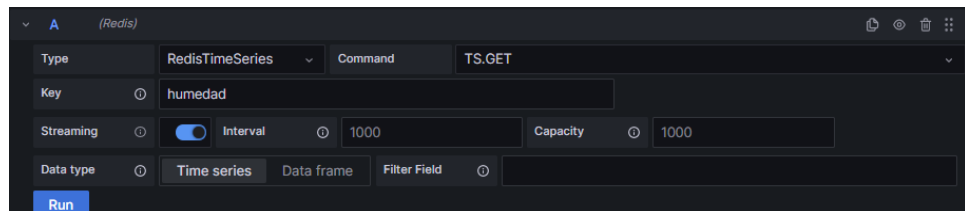


Figura 45. Time Series Humedad.

Por ultimo en este Row, se realiza en una misma grafica la temperatura y humedad, para esto se usó el botón + *Query*, y se realizó de la misma forma que en las gráficas anteriores.

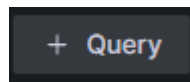


Figura 46. Add Query.

Para el Row Nivel de Agua, se graficó usando el Time Series de Grafana, y especificando la key distancia.

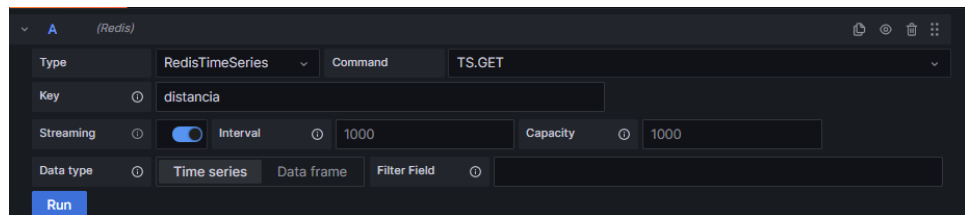


Figura 47. Time Series Nivel del Agua.

En el último Row, esta vez se selecciona el panel de visualización Table y se declaran los datos de la misma forma que en los anteriores pasos.



Figura 48. Selección del panel.

8. Análisis de resultados

8.1. Recopilación de datos y análisis de sensores y actuadores

Para la recopilación de datos se realiza el montaje donde se pueden recibir los datos de la temperatura, humedad y distancia.

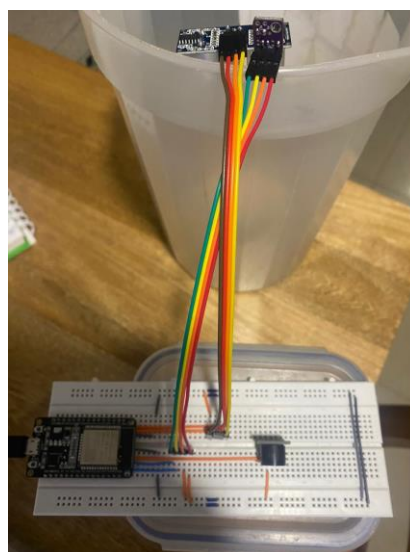


Figura 49. Montaje.

Y los datos se pueden visualizar desde el monitor serie en Arduino.

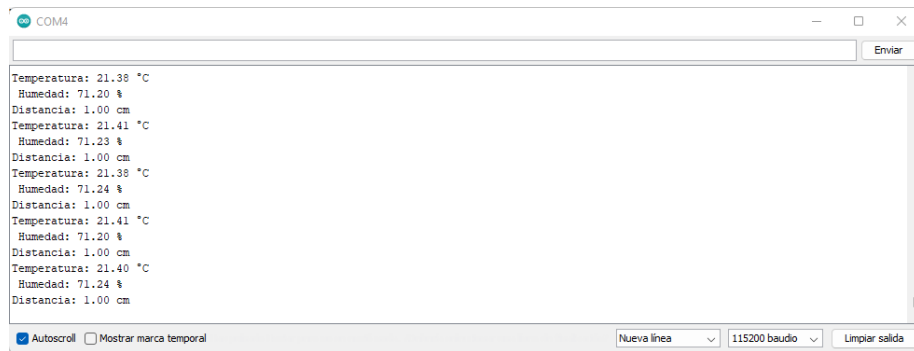


Figura 50. Datos recibidos en la ESP32.

8.2. Envío de datos a treves de comunicación wifi

En Node Red con los datos recopilados mediante la ESP32, se realizó un dashboard en el cual se visualiza los datos de temperatura, humedad y distancia.

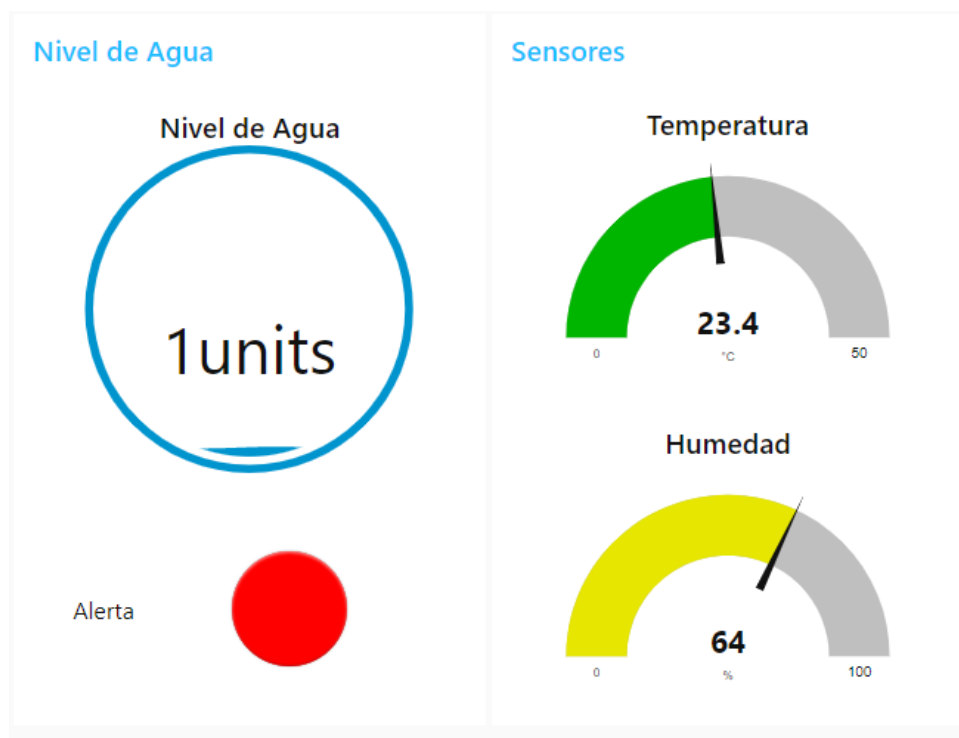


Figura 51. Dashboard en Node Red.

Para el nivel de agua se usó también un led como alerta visual para que luego de cierto nivel en donde ya ha subido mucho el agua, el led se encienda.

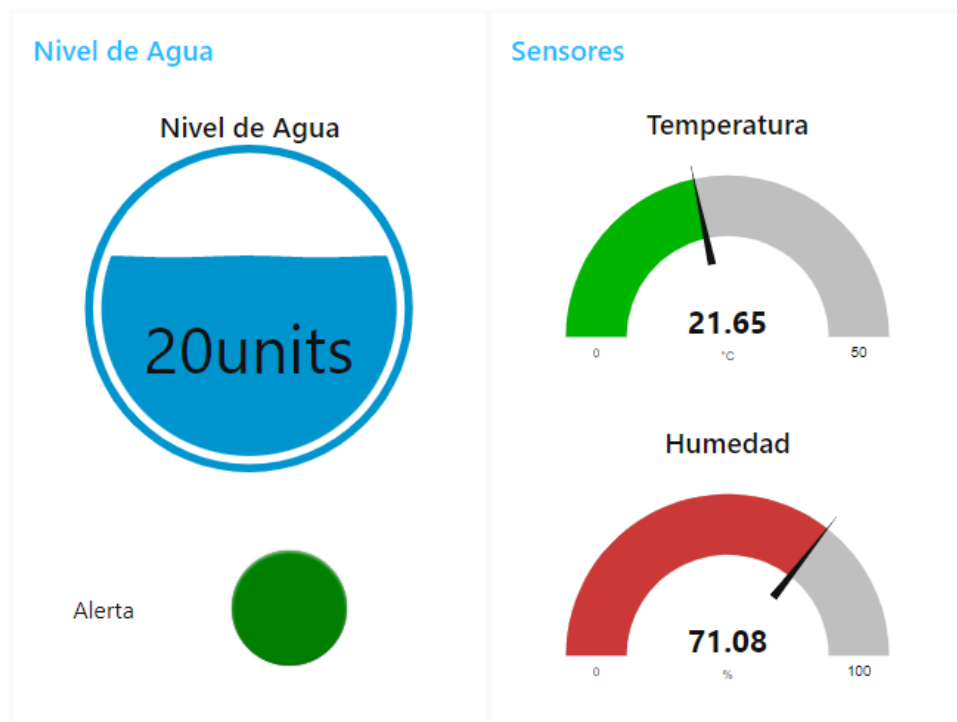


Figura 52. Dashboard con alerta activa.

8.3. Recepción de datos atreves de la Raspberry

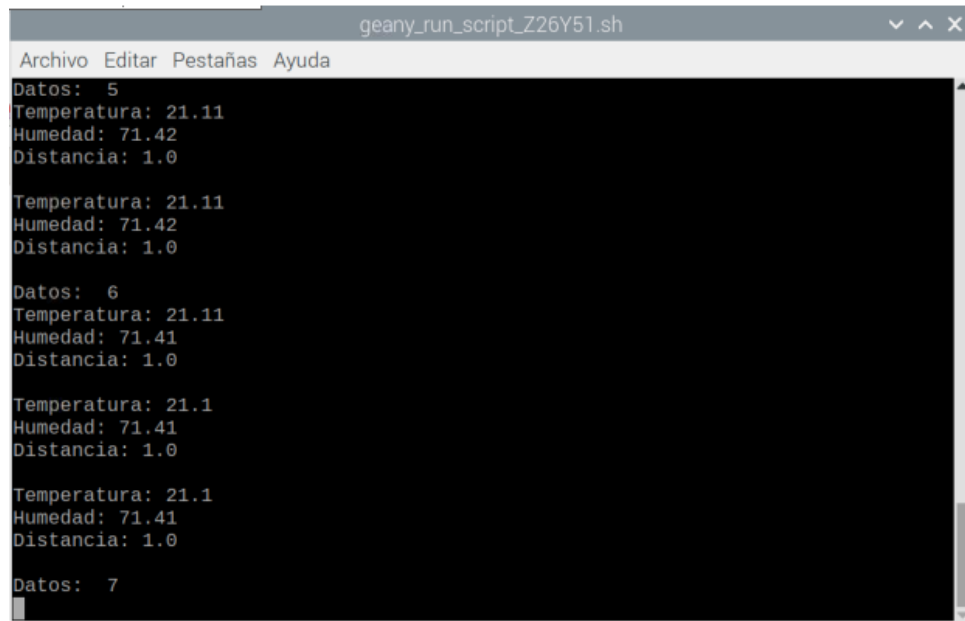
En la Raspberry se programó para crear, cada una de las keys requeridas en el proyecto, estas se pueden visualizar desde Redis Insight.

The screenshot shows the Redis Insight interface with a table of keys. The table has four columns: a type column (all 'TS'), a key name column, a limit column (all 'No limit'), and a size column. There are three keys listed: 'humedad' (49 KB), 'temperatura' (45 KB), and 'distancia' (16 KB). The 'distancia' row is highlighted.

	Filter by Key Nam		+ Key
Total: 3	now		
TS	humedad	No limit	49 KB
TS	temperatura	No limit	45 KB
TS	distancia	No limit	16 KB

Figura 53. Keys en Redis Insight.

Luego, se reciben todos los datos enviados desde Node Red, se imprimen por consola en Geany, pero también se pueden visualizar que aumentan los samples en Redis Insight

A terminal window titled 'geany_run_script_Z26Y51.sh' with a menu bar (Archivo, Editar, Pestañas, Ayuda). The terminal output shows a script running and receiving data. It prints 'Datos: 5' followed by 'Temperatura: 21.11', 'Humedad: 71.42', and 'Distancia: 1.0'. This is repeated for 'Datos: 6' and 'Datos: 7'.

```
geany_run_script_Z26Y51.sh
Archivo Editar Pestañas Ayuda
Datos: 5
Temperatura: 21.11
Humedad: 71.42
Distancia: 1.0

Temperatura: 21.11
Humedad: 71.42
Distancia: 1.0

Datos: 6
Temperatura: 21.11
Humedad: 71.41
Distancia: 1.0

Temperatura: 21.1
Humedad: 71.41
Distancia: 1.0

Temperatura: 21.1
Humedad: 71.41
Distancia: 1.0

Datos: 7
```

Figura 54. Recepción de datos en la Raspberry.

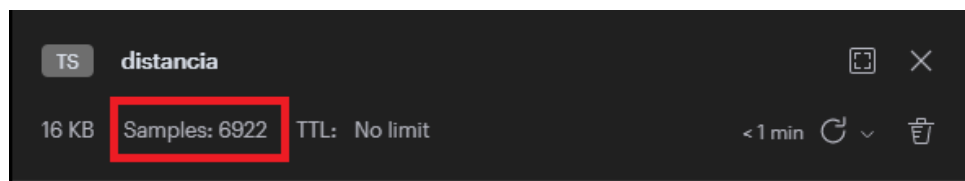


Figura 55. Recepción de datos en Redis.

8.4. Ciencia de datos a partir de Python}

Lo primero que obtuvimos a partir de la ciencia de datos a través de Python fue la creación de archivo csv, este archivo contenía los datos en tiempo real y los guardaba en tiempo real.



Figura 56. Ubicación en carpeta de dato.csv

```

1  datos.csv
2  Fecha,Hora,Temperatura,Nivel_agua,Humedad
3  2023-06-04,13:34:30,26.89837098487192,65.07625373015912,55.79029851884829
4  2023-06-04,13:34:32,29.782447870079263,40.42002383928839,45.49669661578005
5  2023-06-04,13:34:34,26.505685370019503,40.13068324575655,52.999220100416835
6  2023-06-04,13:34:35,21.61142366396681,79.89891007765942,54.84434143170985
7  2023-06-04,13:34:37,27.155951960204945,67.79983641339763,43.849768240627554
8  2023-06-04,13:34:38,24.16252918843394,97.63938428479119,59.85915770025237
9  2023-06-04,13:34:40,20.30565905281454,23.272610388591765,44.14571900949345
10 2023-06-04,13:34:41,23.273858606240204,90.50794041319485,48.32839728496781
11 2023-06-04,13:34:43,22.139379672850993,78.54441134632711,45.49441429664569
12 2023-06-04,13:34:44,25.652465834411863,36.40957311567274,57.533911913667964
13 2023-06-04,13:34:46,24.28301140879037,28.571138231241967,56.50064859400115
14 2023-06-04,13:34:47,20.943911133316572,49.57688531868237,59.58083910402112
15 2023-06-04,13:34:49,23.522275940944873,50.47968929202046,50.04243188857405
16 2023-06-04,13:34:50,28.53136984899549,76.39518563953018,51.444991896300806
17 2023-06-04,13:34:52,23.547754971883474,57.70606120323984,58.85613988478404
18 2023-06-04,13:34:54,21.486926179741094,78.22670033402325,56.954558868970224
19 2023-06-04,13:34:55,24.485367280315465,96.11335597997206,40.1671491883058
20 2023-06-04,13:34:57,20.74456591303389,86.49623677832395,56.586901449743465
21 2023-06-04,13:34:58,22.81670482364936,86.03297558456492,47.751381173346005

```

Figura 57. Datos en carpeta de dato.csv

Luego a partir de la creación del archivo csv se realizó el análisis en tiempo real de los datos que llegan, para hacer la descomposición y poder hallar los patrones de comportamiento que anteriormente se mostraron como son tendencia, estacionalidad y residuos, junto a los datos de temperatura, nivel de agua y humedad.

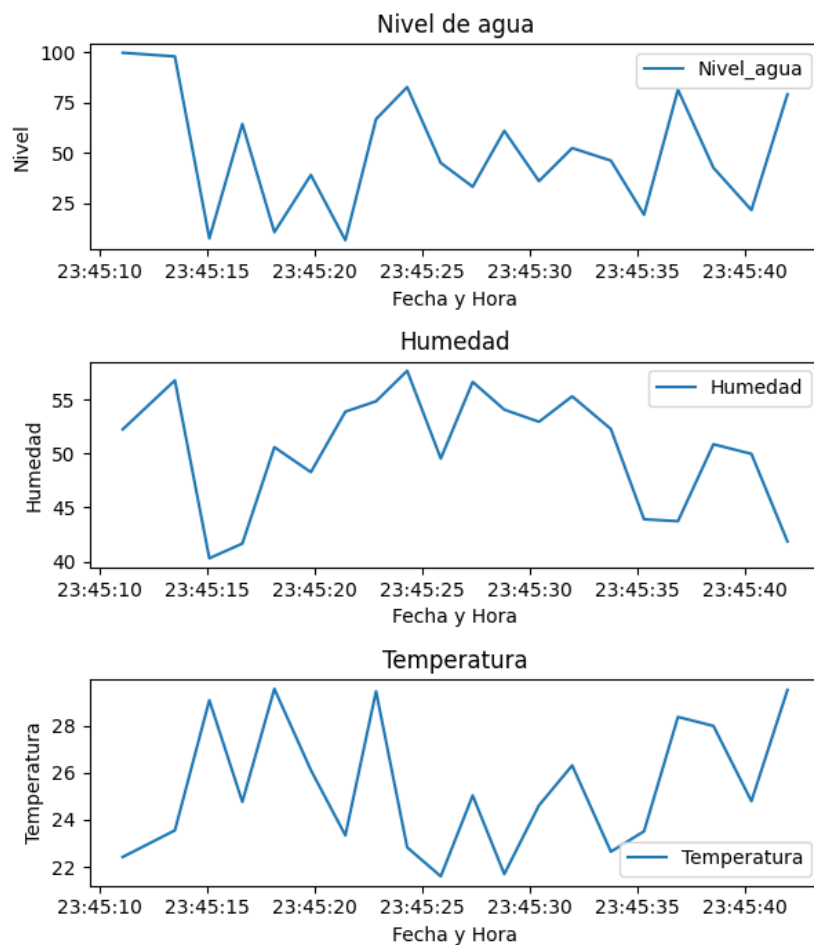


Figura 58. Grafica en tiempo real de las variables de entrada

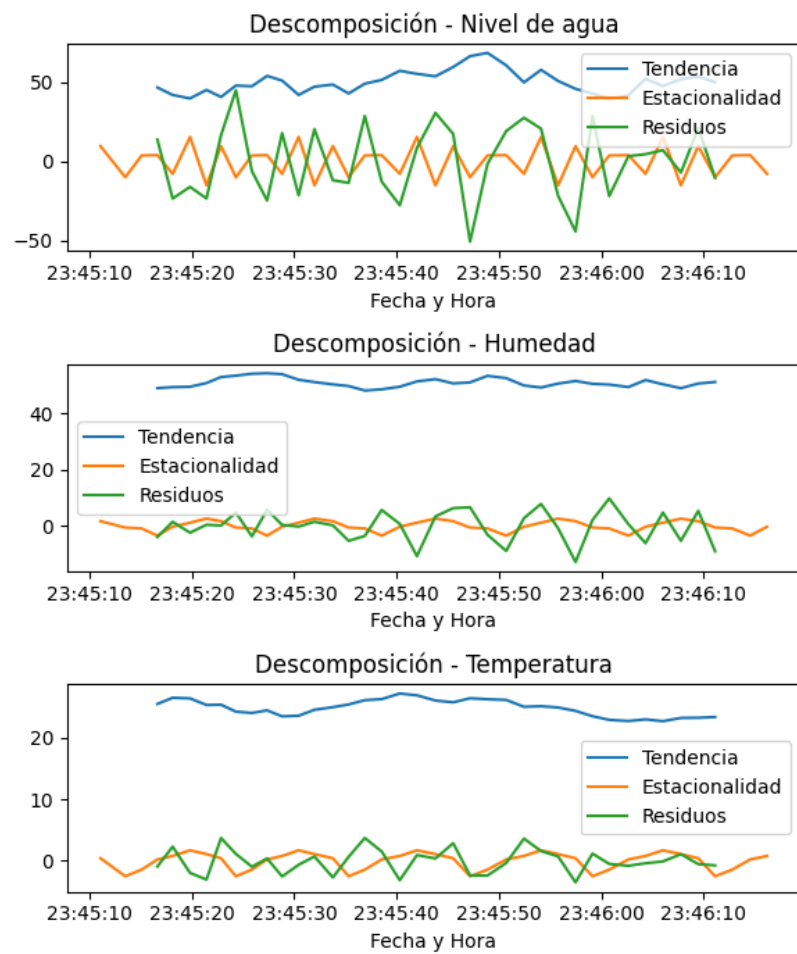


Figura 59. Grafica en tiempo real de la descomposición y análisis del csv

Seguido a esto el usuario podrá interactuar con la recopilación de datos dispuestos en el csv, a partir de la siguiente interfaz de usuario:

Figura 60. Interfaz de usuario

En ella podrá buscar datos mediante fecha y hora o por el nivel del agua, aunque existe una búsqueda avanzada que le permite filtrar información a partir de tres rangos impuestos por el usuario que corresponde a las tres variables.

Figura 61. Frame de búsqueda por Fecha y hora

Resultados:				
Fecha	Hora	Temperatura	Nivel_agua	Humedad
1 2023-06-04	23:45:13	23.550338	97.898643	56.815871

Figura 62. Resultados de búsqueda por Fecha y hora

Buscar por Nivel de Agua

Nivel de agua mínimo: 40

Nivel de agua máximo: 60

Buscar

Figura 63. Frame de búsqueda por rango de nivel de agua

Resultados					
Resultados:					
	Fecha	Hora	Temperatura	Nivel_agua	Humedad
9	2023-06-04	23:45:25	21.594394	45.069049	49.559868
13	2023-06-04	23:45:31	26.315848	52.408105	55.336873
14	2023-06-04	23:45:33	22.643079	46.197799	52.300098
17	2023-06-04	23:45:38	27.999331	42.634748	50.884143
31	2023-06-04	23:46:02	22.100234	48.945149	46.643558
32	2023-06-04	23:46:04	23.345548	48.831334	45.503278
37	2023-06-04	23:46:12	24.626558	51.579708	49.539079
40	2023-06-04	23:46:17	28.308581	52.074464	43.992147
42	2023-06-04	23:46:21	25.523595	55.894501	58.545586
43	2023-06-04	23:46:23	28.339245	51.947627	57.501019

Figura 64. Resultados de búsqueda por rango de nivel de agua

Busqueda Avanzada

Temperatura mínima: 20

Temperatura máxima: 25

Humedad mínima: 40

Humedad máxima: 60

Nivel de agua mínimo: 40

Nivel de agua máximo: 60

Buscar

Figura 65. Frame de búsqueda avanzada

Resultados					
Resultados:					
	Fecha	Hora	Temperatura	Nivel_agua	Humedad
9	2023-06-04	23:45:25	21.594394	45.069049	49.559868
14	2023-06-04	23:45:33	22.643079	46.197799	52.300098
31	2023-06-04	23:46:02	22.100234	48.945149	46.643558
32	2023-06-04	23:46:04	23.345548	48.831334	45.503278
37	2023-06-04	23:46:12	24.626558	51.579708	49.539079

Figura 66. Resultados de búsqueda avanzada

También a la interfaz se le agregara tres ventanas emergentes que mostraran el dato más elevado por columna, pero no solo eso sino también toda la información necesaria para poder tener control directo de las variables.

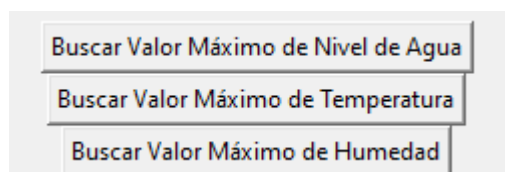


Figura 67. Botones de ventanas emergentes

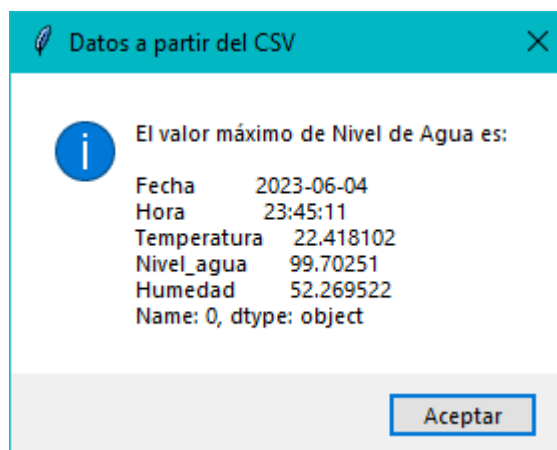


Figura 68. Resultados de búsqueda del mayor Nivel del agua

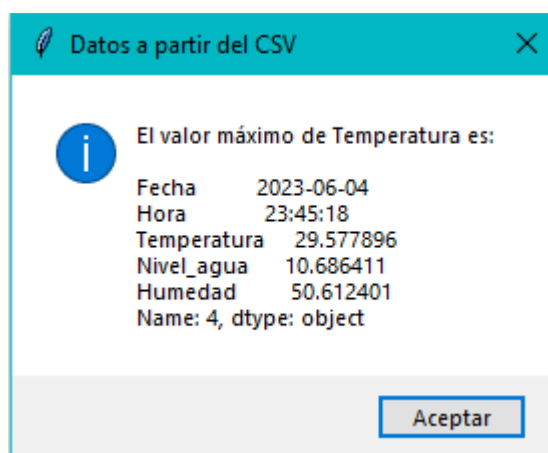


Figura 69. Resultados de búsqueda de la mayor temperatura

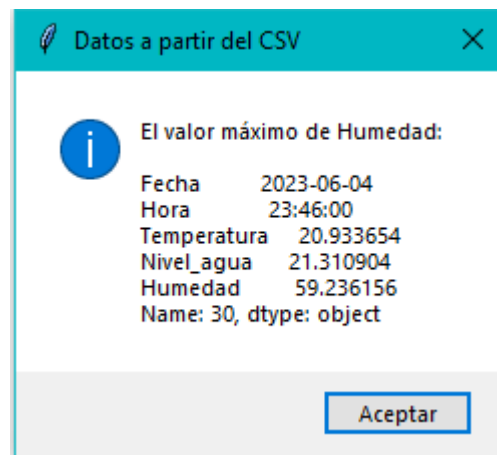


Figura 70. Resultados de búsqueda de la mayor humedad

8.5. Visualización de la información recopilada

Para la visualización en Grafana se usaron 3 Row como se había explicado anteriormente. En el primero se visualizan los datos de la temperatura y la humedad.

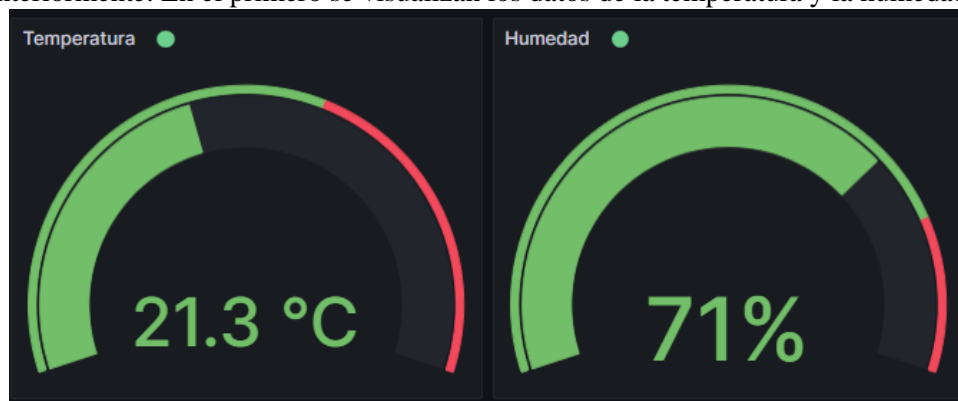


Figura 71. Gauge Temperatura y Humedad.



Figura 72. Grafica Temperatura.

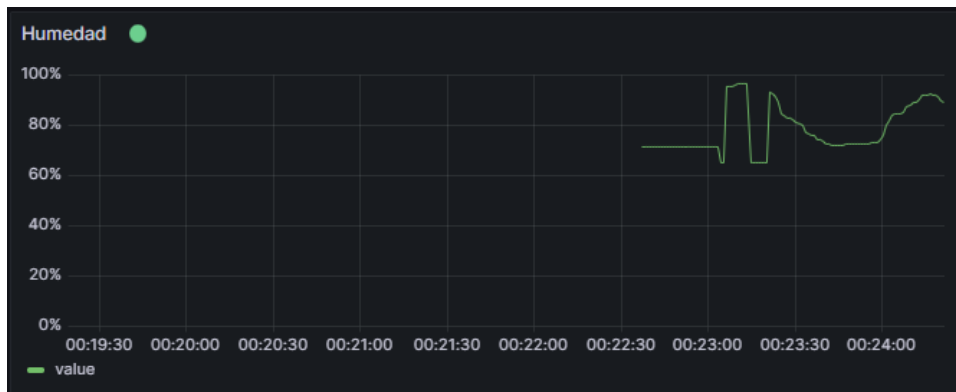


Figura 73. Grafica Humedad.

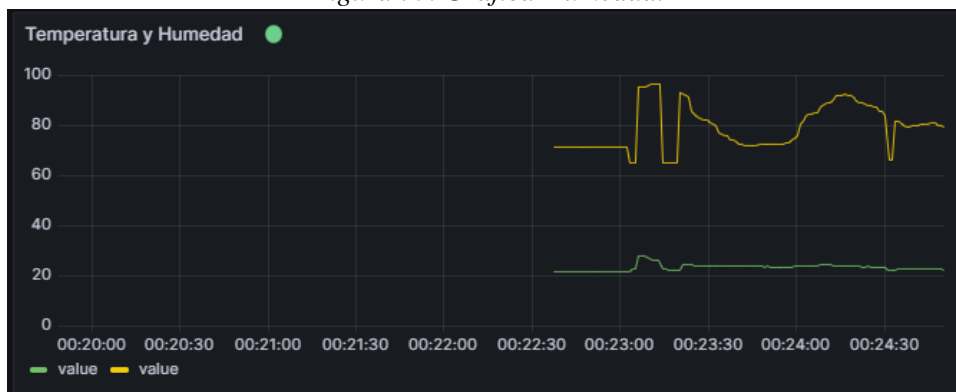


Figura 74. Grafica Temperatura y Humedad.

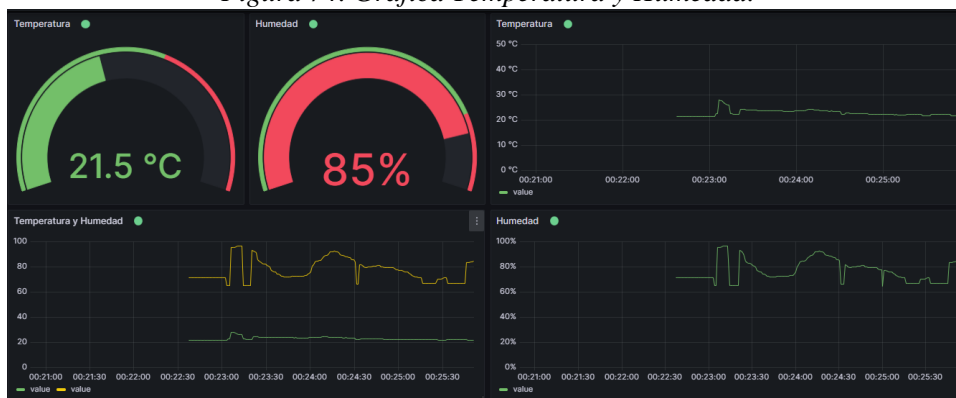


Figura 75. Row Datos de la Temperatura y Humedad.

En el segundo Row, están los datos del nivel del agua.

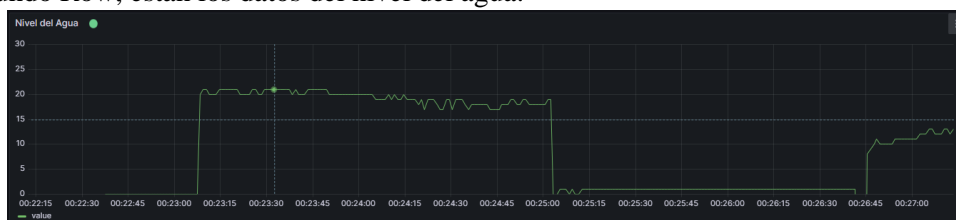


Figura 76. Grafica Nivel del Agua.

Para el ultimo Row, se visualizan los datos en tablas, de las tres variables.

Datos de Temperatura 	
time 	value
2023-06-05 00:22:37	21.3
2023-06-05 00:22:38	21.3
2023-06-05 00:22:38	21.3
2023-06-05 00:22:40	21.3
2023-06-05 00:22:41	21.3
2023-06-05 00:22:42	21.3

Figura 77. Tabla Temperatura.

Datos de Humedad 	
time 	value
2023-06-05 00:22:37	71.3
2023-06-05 00:22:38	71.3
2023-06-05 00:22:39	71.3
2023-06-05 00:22:40	71.3
2023-06-05 00:22:41	71.3
2023-06-05 00:22:42	71.3

Figura 78. Tabla Humedad.

Datos del Nivel del Agua 	
time 	value
2023-06-05 00:22:37	0.0
2023-06-05 00:22:38	0.0
2023-06-05 00:22:39	0.0
2023-06-05 00:22:40	0.0
2023-06-05 00:22:41	0.0
2023-06-05 00:22:42	0.0

Figura 79. Tabla Nivel del Agua.

Datos de Temperatura 		Datos de Humedad 		Datos del Nivel del Agua 	
time 	value 	time 	value 	time 	value
2023-06-05 00:22:37	21.3	2023-06-05 00:22:37	71.3	2023-06-05 00:22:37	0.0
2023-06-05 00:22:38	21.3	2023-06-05 00:22:38	71.3	2023-06-05 00:22:38	0.0
2023-06-05 00:22:38	21.3	2023-06-05 00:22:39	71.3	2023-06-05 00:22:39	0.0
2023-06-05 00:22:40	21.3	2023-06-05 00:22:40	71.3	2023-06-05 00:22:40	0.0
2023-06-05 00:22:41	21.3	2023-06-05 00:22:41	71.3	2023-06-05 00:22:41	0.0
2023-06-05 00:22:42	21.3	2023-06-05 00:22:42	71.3	2023-06-05 00:22:42	0.0

Figura 80. Row Tablas.

9. Conclusiones

En resumen, el desarrollo y diseño de un sistema de alerta temprana de inundaciones utilizando un sensor ultrasónico ha demostrado ser una solución efectiva para detectar y monitorear el nivel del agua en tiempo real. Los resultados de la prueba han sido exitosos, ya que el sensor ultrasónico ha demostrado su capacidad para emitir una alarma cuando se alcanza un nivel de agua crítico.

La utilización del sensor ultrasónico ha brindado una ventaja significativa en la detección de inundaciones, ya que su funcionamiento se basa en la medición precisa de la distancia entre el sensor y la superficie del agua. Esto permite obtener información en tiempo real sobre el nivel del agua, lo que facilita la identificación temprana de posibles inundaciones.

La emisión de una alarma cuando se alcanza un nivel de agua crítico es un factor clave para alertar a las autoridades y a las comunidades sobre la situación de riesgo inminente. Esta función de alerta temprana es fundamental para tomar medidas preventivas y coordinar los esfuerzos de respuesta ante desastres.

La integración del sensor ultrasónico en un sistema de alerta temprana de inundaciones ofrece beneficios adicionales, como la capacidad de monitoreo continuo y la generación de registros de datos. Estos registros de datos pueden ser utilizados para análisis posteriores, detección de patrones, evaluación de riesgos y mejora continua del sistema.

Es importante destacar que el éxito de la prueba del sistema de alerta temprana de inundaciones con el sensor ultrasónico no solo se limita a la detección precisa del nivel del agua, sino también a la efectividad de la alarma emitida. Esto demuestra la confiabilidad y la utilidad del sistema en la protección de las comunidades y la reducción del impacto de los desastres naturales.

Además de la detección y emisión de alarmas, el sistema de alerta temprana de inundaciones basado en el sensor ultrasónico ha permitido la creación y el análisis de una base de datos de registros de nivel de agua. Esta base de datos se ha convertido en una herramienta invaluable para observar y comprender los patrones de comportamiento de las inundaciones.

La recopilación y el almacenamiento de datos en una base de datos ha facilitado el análisis posterior y la identificación de patrones a lo largo del tiempo. Mediante el uso de algoritmos de análisis de datos, se han podido detectar tendencias, fluctuaciones estacionales y anomalías en los niveles de agua. Estos patrones de comportamiento son esenciales para predecir posibles inundaciones y tomar medidas preventivas de manera más precisa y eficiente.

La capacidad de observar y analizar los datos almacenados en la base de datos ha permitido a las autoridades y a los expertos en gestión de desastres obtener una visión más completa de la situación. Esto les ha brindado la posibilidad de tomar decisiones informadas y estratégicas basadas en datos concretos y en la comprensión de los patrones históricos de inundaciones.

Además, la base de datos también ha facilitado la generación de informes y la compartición de información relevante con otras entidades y comunidades afectadas. Esto ha mejorado la coordinación y la colaboración entre diferentes actores involucrados en la gestión de inundaciones, lo que a su vez ha contribuido a una respuesta más eficiente y efectiva ante situaciones de emergencia.

La creación de una base de datos para almacenar y analizar los registros de nivel de agua ha enriquecido el sistema de alerta temprana de inundaciones. Esta capacidad de observar y comprender los patrones de comportamiento de las inundaciones ha mejorado la capacidad de respuesta y la toma de decisiones, proporcionando una base sólida para la planificación y la gestión de desastres a largo plazo.

En conclusión, la implementación de un sistema de alerta temprana de inundaciones basado en un sensor ultrasónico ha demostrado ser una solución eficiente y confiable para la detección temprana de posibles inundaciones. Esta tecnología ofrece una herramienta valiosa para proteger a las comunidades, tomar decisiones informadas y mitigar los riesgos asociados con las inundaciones.

10. Referencias

- Agencia de Protección Ambiental de los Estados Unidos (EPA). (2015). Sistemas de alerta temprana para inundaciones: una herramienta para la planificación y respuesta a inundaciones. EPA.
- Comisión del Río Mekong (MRC). (2019). Sistema de Alerta Temprana de Inundaciones del río Mekong: Visión general y logros. MRC.
- Camargo, F., Rueda, L., & Ordoñez, A. (2021). Diseño e implementación de un sistema de alerta temprana para inundaciones en una cuenca hidrográfica. *Información Tecnológica*, 32(2), 223-230.
- Cedillo, J., Rodríguez, F., Arévalo, J., & Díaz, J. (2019). Sistema de alerta temprana de inundaciones en la cuenca del Río Upano. *Revista Tecnológica - ESPOL*, 32(3), 44-53.
- Cerón, J. D., Torres, H., & Ramos, J. (2020). Diseño e implementación de un sistema de alerta temprana para inundaciones basado en IoT. *Revista Politécnica*, 46(1), 59-64.
- Escamilla, J., Pérez, R., & Hernández, R. (2019). Diseño de un sistema de alerta temprana p
- Deltaprogramma. (2020). El enfoque holandés de gestión del agua: cómo los Países Bajos se mantienen seguros y vivibles en un delta en constante cambio. Deltaprogramma.
- IPCC (Panel Intergubernamental sobre el Cambio Climático). (2014). Cambio climático 2014: Informe de síntesis. Contribución de los Grupos de Trabajo I, II y III al Quinto Informe de Evaluación del IPCC. IPCC.
- IFRC (Federación Internacional de Sociedades de la Cruz Roja y de la Media Luna Roja). (2014). Guía para la participación comunitaria y la rendición de cuentas en la gestión del riesgo de desastres. IFRC.
- UNDRR (Oficina de las Naciones Unidas para la Reducción del Riesgo de Desastres). (2019). Guía de Sendai para la Reducción del Riesgo de Desastres 2015-2030. UNDRR.
- World Bank (Banco Mundial). (2010). Evaluación económica de los desastres naturales en América Latina y el Caribe: Resumen ejecutivo. Banco Mundial.