

Integrantes:

Pachas Saavedra Jesus.

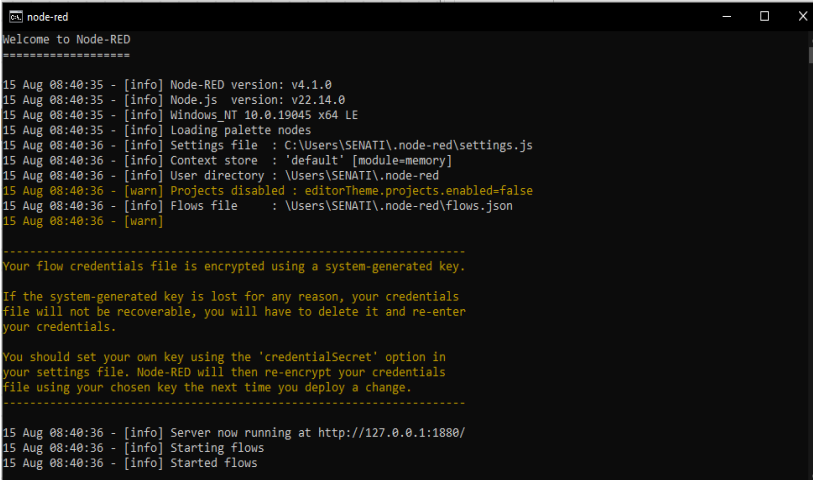
De La Cruz Magallanes Sandra.

## INFORME ARDUINO

En este informe se detallará los procesos realizados en el trabajo correspondiente de Arduino, MongoDB, Y Node-Red

### Proceso:

1. Como primera instancia se tuvo que instalar Node-Red en la Pc para su uso. Luego de instalación de procedió a ejecutarlo en la consola.

A screenshot of a terminal window titled 'node-red'. The window shows the following text: 'Welcome to Node-RED', followed by a series of log messages: '15 Aug 08:40:35 - [info] Node-RED version: v4.1.0', '15 Aug 08:40:35 - [info] Node.js version: v22.14.0', '15 Aug 08:40:35 - [info] Windows\_NT 10.0.19045 x64 LE', '15 Aug 08:40:35 - [info] Loading palette nodes', '15 Aug 08:40:36 - [info] Settings file : C:\Users\SENATI\.node-red\settings.js', '15 Aug 08:40:36 - [info] Context store : 'default' [module-memory]', '15 Aug 08:40:36 - [info] User directory : \Users\SENATI\.node-red', '15 Aug 08:40:36 - [warn] Projects disabled : editorTheme.projects.enabled=false', '15 Aug 08:40:36 - [info] Flows file : \Users\SENATI\.node-red\flows.json', '15 Aug 08:40:36 - [warn]'. Then a block of text explaining flow credentials encryption: 'Your flow credentials file is encrypted using a system-generated key. If the system-generated key is lost for any reason, your credentials file will not be recoverable, you will have to delete it and re-enter your credentials. You should set your own key using the 'credentialSecret' option in your settings file. Node-RED will then re-encrypt your credentials file using your chosen key the next time you deploy a change.' Finally, more log messages: '15 Aug 08:40:36 - [info] Server now running at http://127.0.0.1:1880/', '15 Aug 08:40:36 - [info] Starting flows', and '15 Aug 08:40:36 - [info] Started flows'.

2. Luego se paso a diseñar el código, para poder realizar la prueba Correspondiente.

```
const int trigger = 2;  
const int echo = 3;  
bool deteccion = false;
```

### Definición de Variables:

- **Trigger:** Pin digital 2, conectado al pin Trigger del sensor ultrasónico.
- **Echo:** Pin digital 3, conectado al pin Echo del sensor ultrasónico.
- **Detección:** Variable booleana usada para controlar cuándo se emite una alerta, evitando que se repita constantemente.

```
void setup() {
  Serial.begin(9600);
  pinMode(trigger, OUTPUT);
  pinMode(echo, INPUT);
  digitalWrite(trigger, LOW);
}
```

#### Void Setup:

- Inicializa la comunicación serial a 9600 baudios (para mostrar mensajes en el monitor serial).
- Define los pines: trigger como salida y echo como entrada.
- Asegura que el pin trigger comience en bajo (LOW).

Estas variables almacenan:

- **t**: el tiempo (en microsegundos) que tarda en volver el eco del ultrasonido.
- **d**: la distancia estimada en centímetros.

```
void loop() {
  long t;
  long d;
```

```
digitalWrite(trigger, HIGH);
delayMicroseconds(10);
digitalWrite(trigger, LOW);
```

- Se envía un pulso de 10 microsegundos desde el pin trigger, lo cual activa el sensor ultrasónico para que emita una onda sonora.

- **pulseIn** mide cuánto tiempo tarda en llegar la señal reflejada. Se divide entre 59 para convertir ese tiempo en centímetros (esta es una aproximación común, aunque algunos usan 58).

```
t = pulseIn(echo, HIGH);
d = t / 59;
```

```
if (d < 20 && !deteccion) {
  Serial.println("alerta");
  deteccion = true;
} else if (d > 20 && deteccion) {
  deteccion = false;
}
```

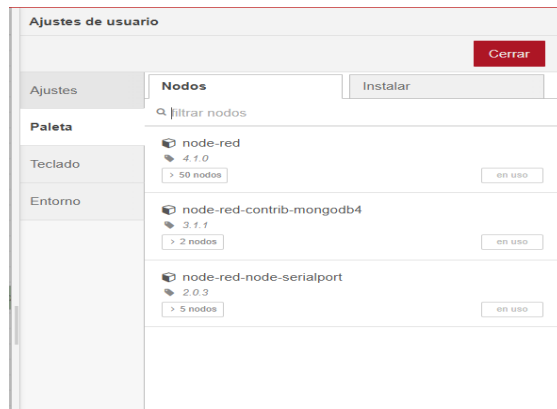
- Si la distancia es menor a 20 cm y aún no se ha detectado nada, se imprime "alerta" y se marca que ya se detectó algo (detección = true).
- Si la distancia vuelve a ser mayor a 20 cm, se reinicia la bandera para permitir futuras detecciones.

- Se espera 100 ms antes de repetir el ciclo.

```
delay(100);
}
```

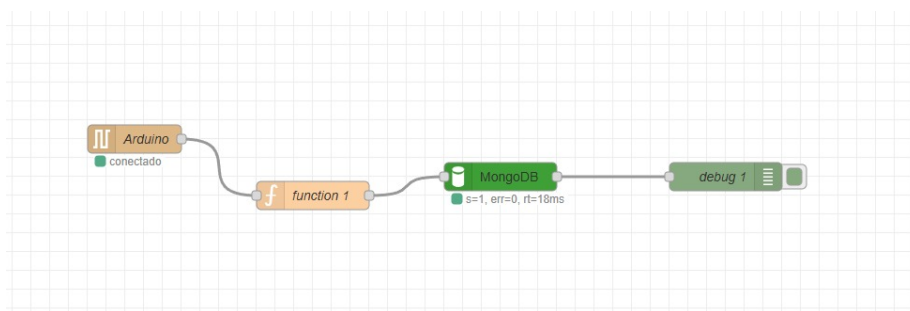
Una Vez probado el código y que haya ejecutado en el software Arduino se pasara hacer la prueba en Node-Red conectado con el MongoDB Y el Arduino.

Como primeras instancias se tuvo que configurar la herramienta Node-RED. Instalando los siguientes puntos.

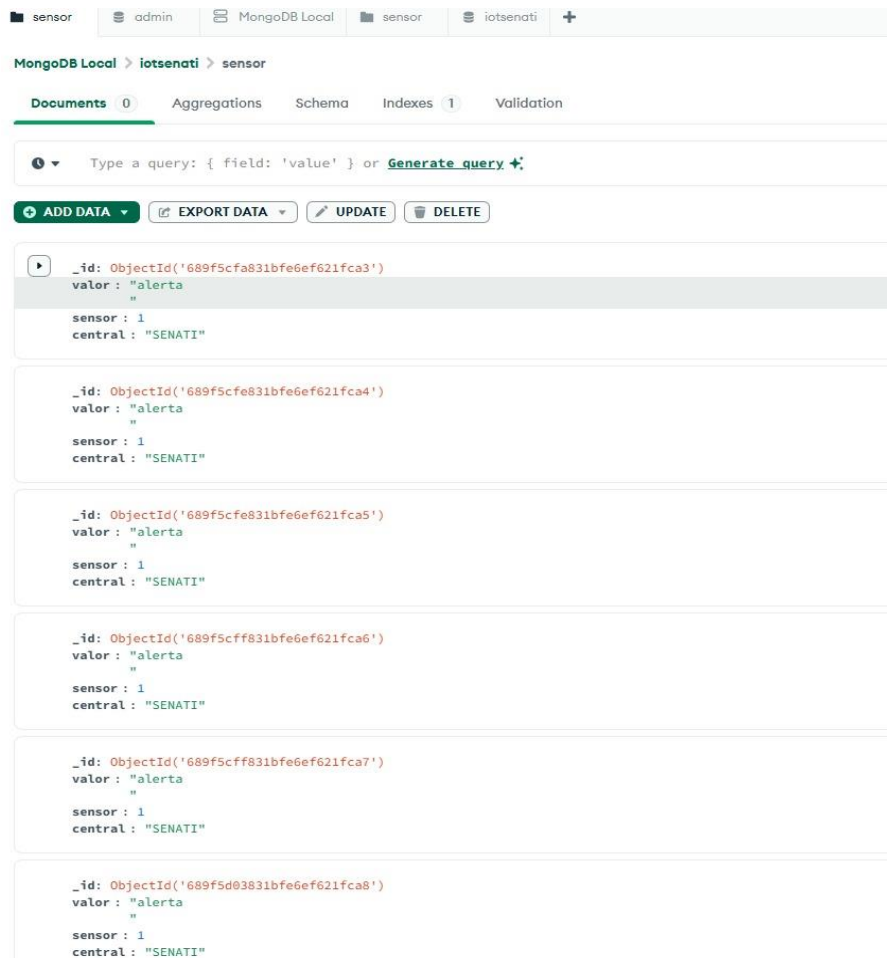


Cuando se tenga instalados en este caso las dependencias, se procederá a poner ir uniendo los procedimientos a utilizar

- **Serial in:** Lee datos de un puerto serie local.
- **Function:** Una función de JavaScript que se ejecuta en los mensajes que recibe el nodo.
- **Mongodb4:** Ejecute operaciones de recopilación de MongoDB con este nodo.
- **Debug:** Muestra las propiedades del mensaje seleccionado en la pestaña de la barra lateral de depuración y, opcionalmente, el registro de tiempo de ejecución.



- Teniendo Los resultados en mongoDB



- Y también saliendo la información en Arduino IDE.

